

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2010

Bc. Karel Ptáček

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

**Softwarová podpora datových formátů používaných pro
měření v elektrotechnice**
**Software Support of Data Formats used for Measurement
in Electrical Engineering Area**

2010

Bc. Karel Ptáček

Zadání diplomové práce

Student: **Bc. Karel Ptáček**
Studijní program: **N2649 Elektrotechnika**
Studijní obor: **2601T004 Měřicí a řídicí technika**
Téma: **Softwarová podpora datových formátů používaných pro měření v elektrotechnice**
Software Support of Data Formats used for Measurement in Electrical Engineering Area

Zásady pro vypracování:

1. Seznámení se s nejpoužívanějšími druhy datových formátů používaných pro měření v elektrotechnice.
2. Seznámení se s technologií Data-Plugin pro softwarový nástroj NI DIAdem.
3. Návrh a vývoj SW podpory pro práci s dosud nepodporovanými datovými formáty v nástroji DIAdem.
4. Ověření funkčnosti aplikace a zhodnocení použité technologie.

Seznam doporučené odborné literatury:

1. *LabVIEW Fundamentals* [online]. Austin: National Instruments, 08/2005 [cit.2009-09-09]. Dostupné z: < <http://www.ni.com/pdf/manuals/374029a.pdf> >. 165s.
2. *LabVIEW FPGA module User Manual* [online]. Austin: National Instruments, 03/2004 [cit.2009-09-09]. Dostupné z: < <http://www.ni.com/pdf/manuals/374690b.pdf> >. 62s.
3. ŽÍDEK, J. *Virtuální instrumentace na bázi grafického programování*. Ostrava: VŠB-TU Ostrava, 2003. 125s. ISBN 80-248-0298-8.
4. HAVLÍČEK, J.; VLACH, J.; VLACH, M.; VLACHOVÁ, V. *Začínáme s LabVIEW*. Praha: BEN, 2008. 248 s. ISBN 978-80-7300-245-9.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Karel Ptáček

Datum odevzdání diplomové práce : 7.5.2010

Poděkování

Chtěl bych touto cestou velmi poděkovat vedoucímu mé diplomové práce panu doc. Ing. Petru Bilíkovi, Ph.D za cenné rady, konzultace a připomínky spojené s vypracováním mého úkolu.

Abstrakt

Cílem této diplomové práce je vytvoření softwarové podpory datových formátů, které jsou používány v elektrotechnice. Takových to formátů je dnes několik desítek. Práce je konkrétně zaměřena na podporu datových formátů COMTRADE a formátů, které využívá společnost ELCOM. Podpora je vytvářena v LabVIEW pomocí DataPluginu, který bude zajišťovat konverzi dat nepodporovaných formátů do formátů podporovaných softwarem DIAdem.

Výsledky této práce mohou posloužit pro analyzování zmíněných datových formátů právě v prostředí DIAdem.

Klíčová slova

datové formáty, COMTRADE, DataPlugin, LabVIEW, DIAdem, TDMS

Abstract

Target of this thesis is to create software support for the data formats which are used in the electrical engineering area. There are a few tens of this formats. The work is focused on the COMTRADE data formats and the formats which are used by the ELCOM company. Support is created in the LabVIEW using the DataPlugin, which will ensure data conversion from unsupported formats into formats which are supported by the DIAdem software.

Results of this work can be used for analyzing mentioned data formats right in the DIAdem environment.

Key words

data formats, COMTRADE, DataPlugin, LabVIEW, DIAdem, TDMS

Seznam použitých symbolů a zkratek

ASCII	American Standard Code for Information Interchange	znaková sada
COMTRADE	Common Format for Transient Data Exchange	datový formát
LSB	Least Significant Bit	nejméně významný bit
MSB	Most Significant Bit	nejvíce významný bit
NI	National Instruments	mezinárodní společnost
POW		proprietární datový formát analyzátoru sítí
SW	Software	programové vybavení
TDM	Technical Data Management	datový formát
TDMS	Technical Data Management Streaming	datový formát
TRA		proprietární datový formát analyzátoru sítí
VI	Virtual Instrument	program v LabVIEW
XML	Extensible Markup Language	standardní formát pro výměnu informací

Obsah

1	Úvod	- 1 -
2	Nejpoužívanější druhy datových formátů pro měření v elektrotechnice	- 2 -
2.1	TDM – Technical Data Management	- 2 -
2.2	TDMS – Technical Data Management Streaming	- 4 -
2.3	COMTRADE	- 5 -
2.3.1	Hlavičkový soubor	- 5 -
2.3.2	Konfigurační soubor	- 6 -
2.3.3	Datový soubor	- 7 -
2.3.4	Informační soubor	- 9 -
3	DIAdem	- 10 -
3.1	DIAdem DataFinder	- 11 -
4	DataPlugin	- 12 -
4.1	Typy DataPluginu	- 12 -
4.2	Vytváření DataPluginu	- 13 -
4.3	Testování a ladění DataPluginu	- 14 -
5	DataPlugin one shot	- 15 -
5.1	DataPlugin proprietárního datového formátu	- 15 -
5.1.1	Struktura souboru data.xyz:	- 15 -
5.1.2	Návrh jádra DataPluginu	- 15 -
5.1.3	Testování	- 17 -
5.1.4	Vytváření knihovny DataPluginu	- 19 -
5.1.5	Vytváření instalátoru DataPluginu	- 21 -
5.1.6	Instalace DataPluginu do systému	- 23 -
5.1.7	Použití DataPluginu v DIAdemu	- 24 -
5.2	DataPlugin podporující souborový formát COMTRADE	- 26 -
5.2.1	Návrh VI pro zapisování dat ve formátu COMTRADE	- 26 -
5.2.2	Návrh VI pro čtení dat ze souboru COMTRADE	- 27 -
5.2.3	Návrh jádra DataPluginu one shot pro podporu COMTRADE	- 28 -
5.2.4	Testování DataPluginu podporujícího COMTRADE	- 30 -

5.3	DataPlugin podporující souborové formáty TRA	- 34 -
5.3.1	Návrh jádra DataPluginu one shot pro podporu *. TRA	- 34 -
5.3.2	Testování DataPluginu podporujícího formát TRA	- 35 -
5.4	DataPlugin podporující souborové formáty POW	- 38 -
5.4.1	Návrh jádra DataPluginu one shot pro podporu *. POW	- 38 -
5.4.2	Testování DataPluginu podporujícího formát POW	- 39 -
6	DataPlugin on demand podporující formát COMTRADE	- 42 -
6.1	Návrh jádra DataPluginu	- 42 -
6.1.1	Meta data	- 42 -
6.1.2	Raw data	- 43 -
6.2	Testování DataPluginu podporujícího COMTRADE	- 44 -
7	Závěr	- 46 -
	Použitá literatura	- 47 -
	Seznam příloh	- 48 -

Seznam obrázků

Obr. 1 TDM datový model.....	- 2 -
Obr. 2 Struktura souborů v TDM formátu	- 3 -
Obr. 3 Struktura souborů v TDMS formátu	- 4 -
Obr. 4 Pracovní plocha modulu VIEW SW DIAdem	- 10 -
Obr. 5 Seznam DataPluginů zobrazených prostředkem DIAdem DataFinder	- 11 -
Obr. 6 Mapování neznámého formátu s využitím TDM datového modelu	- 12 -
Obr. 7 DataPlugin projekt	- 13 -
Obr. 8 Blokový diagram jádra DataPluginu	- 16 -
Obr. 9 CreateDataPluginOutputFile.vi.....	- 17 -
Obr. 10 Prohlížeč dat ze souboru data1.xyz po spuštění DataPluginu	- 17 -
Obr. 11 Properties a hodnoty kanálu nazvaného Sin.....	- 18 -
Obr. 12 Vytváření knihovny.....	- 19 -
Obr. 13 Konfigurace <i>RegisterLabVIEWDataPlugin.ini</i>	- 19 -
Obr. 14 Nastavení názvu DataPluginu a cílového adresáře	- 20 -
Obr. 15 Version information	- 20 -
Obr. 16 Tlačítko <i>Build</i>	- 20 -
Obr. 17 Vytváření instalátoru.....	- 21 -
Obr. 18 Nastavení názvu DataPluginu a cílového adresáře	- 21 -
Obr. 19 Nastavení zdrojového souboru	- 22 -
Obr. 20 Tlačítko <i>Build</i>	- 22 -
Obr. 21 Obsah instalačního adresáře DataPluginu	- 23 -
Obr. 22 Instalace DataPluginu <i>DataPluginASCIIxyz</i>	- 23 -
Obr. 23 Podpora souborového formátu *.xyz v DIAdemu	- 24 -
Obr. 24 Načtení souboru data1.xyz	- 25 -
Obr. 25 Obsah souboru data1.xyz rozdělený do jednotlivých kanálů	- 25 -
Obr. 26 Blokový diagram čtení dat z COMTRADE souboru	- 26 -
Obr. 27 Zápis properties (vlastností) digitálních kanálů	- 27 -
Obr. 28 Čtení properties souboru	- 28 -
Obr. 29 Blokový diagram DataPluginu COMTRADE	- 29 -
Obr. 30 Properties analogového kanálu	- 30 -
Obr. 31 Hodnoty digitálního kanálu zachycené v grafu.....	- 30 -
Obr. 32 Hodnoty analogového kanálu zachycené v grafu.....	- 31 -
Obr. 33 Načtené skupiny a kanály do datového portálu DIAdemu.....	- 32 -
Obr. 34 Tabulka s načtenými daty z kanálů v DIAdemu	- 32 -
Obr. 35 Graf analogového a digitálního kanálu zobrazeného v sekundách v DIAdemu	- 33 -
Obr. 36 Blokový diagram DataPluginu TRA.....	- 34 -
Obr. 37 Properties souboru.....	- 35 -
Obr. 38 Hodnoty kanálu L1 (skupina U).....	- 35 -
Obr. 39 Hodnoty kanálu L1 zachycené v grafu (skupina U).....	- 36 -
Obr. 40 Graf kanálu L1 (skupina U) zobrazeného v sekundách v DIAdemu	- 37 -
Obr. 41 Blokový diagram DataPluginu POW	- 38 -
Obr. 42 Properties souboru.....	- 39 -
Obr. 43 Hodnoty kanálu THDu (skupina L1)	- 39 -
Obr. 44 Hodnoty kanálu THDu zachycené v grafu (skupina L1)	- 40 -

Obr. 45 Graf kanálu THDu (skupina L1) zobrazený v sekundách v DIAdemu	- 41 -
Obr. 46 Princip funkce DataPluginu typu on demand.....	- 42 -
Obr. 47 Blokový diagram části jádra s META daty	- 43 -
Obr. 48 Hodnoty analogového kanálu zachycené v grafu.....	- 44 -
Obr. 49 Graf analogového kanálu zobrazeného v sekundách v DIAdemu	- 45 -

1 Úvod

V dnešní době je již většina měření v elektrotechnice prováděna digitální technikou, což přináší mnoho výhod, ale samozřejmě i některé nevýhody. V současnosti existuje mnoho souborových formátů, podle nichž mohou být ukládány data. Každý z těchto formátů ukládá data jiným způsobem do jiných dílčích souborů, které jsou daným typem formátu definované. Často lze říci, co výrobce, to jiný formát. Pak dochází k tomu, že data uložená softwarem jednoho výrobce nemohou být analyzována v programu jiného výrobce. Ideálním řešením by bylo, kdyby byl vytvořen univerzální standardizovaný souborový formát, který by vyhovoval všem výrobcům. Největší nevýhodou doposud využívaných datových formátů je nekompatibilita s jinými datovými soubory.

V kapitole 2 jsou popsány některé používané datové formáty, jako je souborový formát TDM a TDMS, na nichž je postaven software označovaný jako DIAdem, který je vyvinut společností National Instruments. Dalším často používaným formátem je COMTRADE.

Kapitola 3 popisuje produkt společnosti National Instruments DIAdem, který slouží pro analyzování naměřených dat.

Kapitola 4 se zabývá problematikou vytváření a testování DataPluginu, který je v podstatě určitým převodníkem mezi nepodporovaným datovým formátem a podporovaným datovým formátem. Konkrétně se jedná o nepodporované soubory v softwaru DIAdem.

Kapitola 5 se zabývá vytvářením DataPluginu typu one shot. Nabízí pohled na vytváření konkrétních DataPluginů, a to např. doposud nepodporovaného formátu XYZ, který byl navržen proto, aby na jednoduchém příkladu byl vysvětlen vývoj DataPluginu. V rámci něj je popsán samotný návrh jádra DataPluginu, testování, a v neposlední řadě také i vytváření knihovny a instalátoru tohoto DataPluginu. Dále jsou v této kapitole popsány návrhy DataPluginů podporující souborové formáty COMTRADE, formáty TRA a POW.

Kapitola 6 je zaměřena na vytváření DataPluginu typu on demand pro konkrétní formát COMTRADE. Tak jako v předchozí kapitole je zde opět popsán samotný návrh a testování DataPluginu.

2 Nejpoužívanější druhy datových formátů pro měření v elektrotechnice

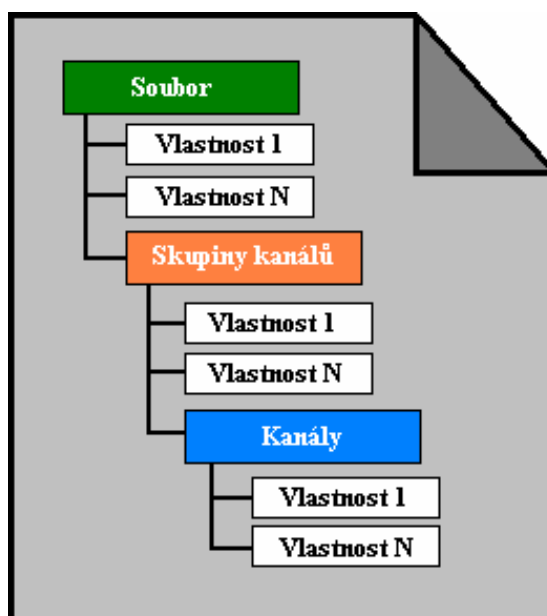
Pro ukládání naměřených dat je v dnešní době možno si vybrat z desítek datových formátů, které se od sebe odlišují svými vlastnostmi, složitostí struktury, ale i svou použitelností v daných oblastech měření. Některé z těchto datových formátů budou dále podrobněji popsány. Jedná se o formáty, které jsou použity v rámci této diplomové práce, a to sice TDM, TDMS, COMTRADE.

2.1 TDM – Technical Data Management

TDM formát pro ukládání dat je relativně nový model navržený firmou National Instrument. Struktura vychází z TDM datového modelu a je složena z definičního bloku a samostatného bloku naměřených dat. Dané měření i ukládaný kanál lze popsat neomezeným počtem parametrů, které může uživatel naplnit libovolnými hodnotami, ovšem pouze jednoduchého datového typu. Popis měřených kanálů lze uložit ve formátu XML do samostatného souboru. Samostatná data jsou uložena do dalšího odděleného souboru. Měření je tvořeno dvěma soubory. Ukládat lze pouze data jednoduchého datového typu (každý kanál má přiřazenu datovou strukturu jednorozměrného pole dat), není zde možné ukládat data složitější struktury, vícerozměrná pole a podobné složitější datové typy.

TDM datový model je hierarchický třístupňový datový model obsahující tyto tři úrovně:

- Základní úroveň File (Root) – sekce určená pro popis vlastností příslušejících celému souboru
- Úroveň skupin Groups – sekce určená pro popis vlastností skupin kanálů
- Úroveň kanálů Channels – sekce určená pro popis vlastností jednotlivých kanálů

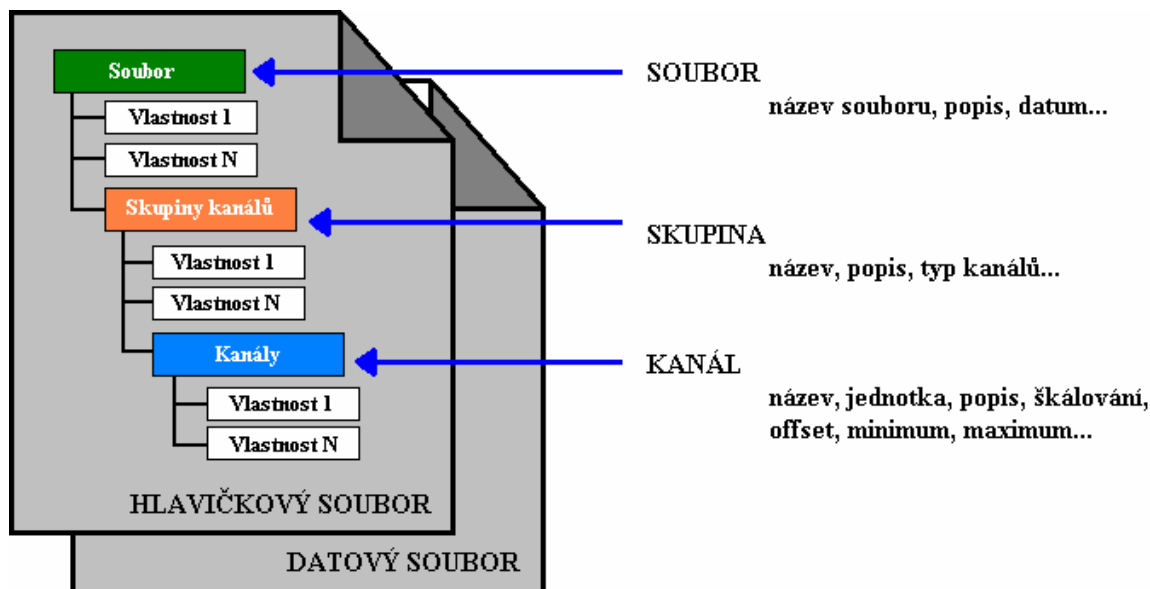


Obr. 1 TDM datový model

Datový formát TDM je založen na XML standardu. Rozkládá se při ukládání informací do dvou samostatných souborů:

- Hlavičkový soubor obsahuje popisné informace o měřených datech (*.tdm)
- Datový soubor obsahuje měřená data (*.tdx)

Hlavičkový soubor je využíván SW prostředky managementu naměřených dat k vyhledání odpovídajících dat.



Obr. 2 Struktura souborů v TDM formátu

Uložená data v TDM datovém formátu je možné zobrazit i v tabulkovém procesoru Excel. Aby to bylo možné, je potřeba nainstalovat doplněk *TDM Excel Add-in*, který rozšíří Excel o nabídku *Import a TDM File*. Po importu měřených dat vzniknou v sešitu Excel dva listy. Na prvním listu je obsah hlavičkového souboru a na druhém listu jsou importována binární naměřená data.

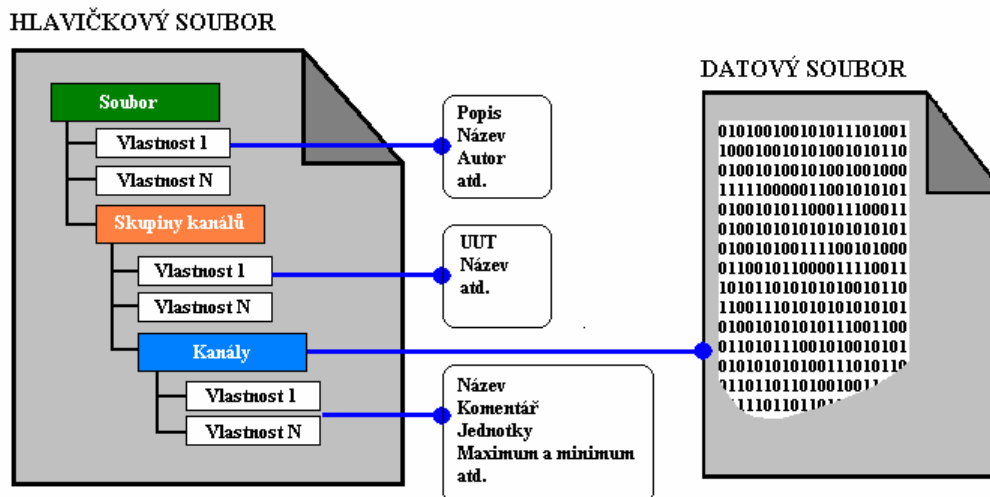
Doplněk je ke stažení na stránce <http://zone.ni.com/devzone/cda/epd/p/id/2944>.

Na adrese <http://zone.ni.com/devzone/cda/tut/p/id/4906> lze nalézt návod k použití tohoto doplňku tabulkového procesoru Excel.

[1]

2.2 TDMS – Technical Data Management Streaming

Naměřená data uložená ve formátu TDMS jsou druhým způsobem mapování TDM datového modelu. Tento datový formát obsahuje, narozdíl od TDM formátu, pouze jediný soubor, a to binární soubor, který slouží zároveň jak pro hlavičková data, tak i pro data naměřená.



Obr. 3 Struktura souborů v TDMS formátu

Každý TDMS soubor psaný v LabVIEW obsahuje popisné informace (properties) o všech třech úrovních. TDMS soubor poskytuje souhrnné informace o všech atributech a ukazatelích naměřených dat a zrychluje přístup k datům během čtení. Zapisování do souborů formátu TDMS se často používá pro práci se systémy reálného času (real-time systems), kde jsou měřená data velmi rychle ukládána na deterministický systém. Soubor může obsahovat nekonečný počet skupin a každá skupina nekonečný počet kanálů. Typické datové typy v TDMS souborech jsou integer, floating point, časové razítko a string. Pole nebo komplexní datové typy nejsou v tomto formátu podporované. Podporované jsou pouze jednorozměrné pole.

TDMS soubory obsahují dvojí data: Meta data a RAW data. Meta data jsou popisové data uložené v objektech nebo vlastnostech. Datová pole připojená ke kanálu jsou označována jako RAW data. Data jsou do TDMS souboru zapisována v segmentech. Pokaždé, když se připojí data do souboru, tak se vytvoří nový segment. Každý segment začíná TDMS tagem, který zabraňuje duplicitě dat v TDMS souboru.

[2]

2.3 COMTRADE

Tento standard definuje formát pro soubory, které zachycují přechodový průběh a události dat výkonových systémů. Tento formát je zamýšlen pro poskytnutí jednoduché interpretace výměny dat. Je to standard, který je určen pro ukládání veškerých dat na fyzické média nebo na digitální diskové jednotky. Není to standard použitelný pro přenos dat mezi komunikačními sítěmi. Tento standard se tedy používá pro měření dat v elektrotechnice z výkonových systémů, ale je možné jej použít i v oblasti ochrany v elektrotechnice. Jedná se o využití tohoto formátu v případě, že dojde během činnosti zařízení k nějaké poruše, jejíž průběh (popis poruchy) je zaznamenán v COMTRADE souboru.

Každý COMTRADE záznam se skládá ze čtyř od sebe oddělených souborů. Každý z nich se stará o informace jiným způsobem. Jsou to tyto následující soubory:

- Hlavičkový
- Konfigurační
- Datový
- Informační

Všechny tyto soubory musí mít stejné jméno, liší se pouze příponou indikující o jaký soubor se jedná. Hlavičkový soubor má příponu .HDR, konfigurační .CFG, datový .DAT a informační .INF.

2.3.1 Hlavičkový soubor

Hlavičkový soubor je ASCII textový soubor doplněný informacemi nabízející uživateli lepší porozumění záznamů. Není určen pro manipulaci aplikačními programy.

Příkladem informací v hlavičkovém souboru může být:

- Název měřicí stanice (např. PC)
- Identifikace jednotlivých zařízení, jakož to mohou být kanály, kondenzátory, reaktory, transformátory apod.
- Vzájemné propojení mezi paralelními kanály
- Umístění a zatížitelnost odstaveného reaktoru
- Jmenovitá napěťová zatížitelnost transformátorového vinutí

[3]

2.3.2 Konfigurační soubor

Konfigurační soubor je ASCII textový soubor, který poskytuje informace potřebné pro člověka nebo počítačový program ke čtení a interpretaci datových hodnot v oddělených datových souborech. Konfigurační soubor je v předdefinovaném, standardizovaném formátu tak, aby se počítačový program nemusel přizpůsobovat každému konfiguračnímu souboru.

Konfigurační soubor obsahuje následující informace:

- Název měřicí stanice, identifikace zařízení pořizující záznam a roku revize COMTRADE standardu
- Celkový počet kanálů, počet analogových kanálů, počet digitálních kanálů
- Parametry analogových kanálů
 - Číslo kanálu, název kanálu, fázová identifikace kanálu, okruh komponent pro monitorování, jednotky, kanálová násobička, offset, čas kanálu od startu vzorkovací periody, minimální a maximální hodnota kanálu, primární a sekundární faktor napětových nebo proudových kanálů
- Parametry digitálních kanálů
 - Číslo kanálu, název kanálu, fázová identifikace kanálu, okruh komponent pro monitorování, normální stav kanálu (buď 0, nebo 1)
- Kanálová frekvence - jmenovitá frekvence v Hz
- Vzorkovací perioda a počet vzorků na každou periodu
- Datum a čas první datové hodnoty
- Datum a čas trigrovacího bodu
- Typ datového souboru (ASCII nebo BINARY)
- Časové razítko násobného faktoru

Celkový počet kanálů, tzn. analogových i digitálních dohromady, může být maximálně 999 999. Analogových kanálů může být maximálně 999 999. Digitálních kanálů může být maximálně 999 999.

[3]

Zobrazená struktura konfiguračního souboru je v příloze I.

2.3.3 Datový soubor

Datový soubor obsahuje datové hodnoty. Data musí přesně odpovídat danému datovému formátu, který je definován v konfiguračním souboru. Dále podle toho, jak je v konfiguračním souboru nadefinován typ souboru (ASCII nebo BINARY), je pak reprezentován datový soubor ve čtečce.

Zapsaná data v datovém souboru se skládají z čísla vzorku, časového razítka a datových hodnot každého kanálu každého vzorku v souboru. Pokud se jedná o datový soubor zapsaný v ASCII, pak jsou jednotlivé hodnoty od sebe odděleny čárkou, pokud se jedná o binární reprezentaci datového souboru, pak se jednotlivé hodnoty mezi sebou neoddělují čárkou, není zde žádný oddělovač.

Datové soubory *.dat jsou omezeny, velikostí 3,5 palcové diskety, tedy velikostí 1,44MB z historických důvodů, data mohou být uložena až ve 100 datových souborech (*.D00 - *.D99), kde každý takový soubor může obsahovat maximálně 10 miliard řádků (9 999 999 999 záznamů).

Datový soubor v ASCII formátu

Hodnoty analogových kanálů mohou být pouze integer, minimální hodnota je -99 999 a maximální hodnota 99 998. Chybějící analogová hodnota je reprezentovaná v poli hodnotou 99 999. Hodnoty digitálních kanálů jsou integer, ale mohou nabývat pouze dvou platných hodnot a to 0 a 1.

Takto reprezentovaný datový soubor se skládá z řádků a ze sloupců. Počet řádků je závislý jednak na délce záznamu, a jednak je ovlivněn i délkou souboru. Každý řádek je rozdělen na TT+2 sloupců, kde TT je celkový počet kanálů v záznamu (analogových i digitálních) a další dva sloupce jsou určeny pro číslo vzorku a pro časové razítko. Počet sloupců je závislý na délce souboru.

Struktura datového souboru:

- První sloupec obsahuje číslo vzorku
- Druhý sloupec je složen z časového razítka pro příslušný vzorek
- Třetí sloupec je složen z tolika dílčích sloupců, kolik je analogových kanálů, a ty obsahují hodnoty reprezentující analogovou informaci
- Čtvrtý sloupec je složen z tolika dílčích sloupců, kolik je digitálních kanálů, a ty obsahují hodnoty reprezentující digitální informaci
- Další řádek opět začíná číslem vzorku a příslušnou datovou sadou
- Jednotlivé sloupce jsou od sebe odděleny čárkou.

Zápis vzorku vypadá následovně: **n,timestamp,A₁,A₂...A_k,D₁,D₂...D_m** ,

kde n je číslo vzorku, timestamp je časové razítko, A₁ až A_k jsou hodnoty jednotlivých analogových kanálů a D₁ až D_m jsou hodnoty jednotlivých digitálních kanálů.

Základní jednotkou časového razítka je μs (microseconds [mikrosekunda]).

Malý příklad ASCII souboru:

```
1,    0, -994, 1205, 100, 29, -135, -197,0,0,0,0,0 <CR/LF>
2,   167, -943, 1231, 94, 37, -137, -275,0,0,0,0,0 <CR/LF>
3,   333, -886, 1251, 87, 45, -139, -351,0,0,0,0,1 <CR/LF>
...
885, 147333, 360, -387, -2, 0, -1, -385,0,0,0,0,0 <CR/LF>
<1A><CR/LF>
```

Datový soubor v binárním formátu

Binární datový soubor používá podobně jednoduchou strukturu jako ASCII datový soubor, obsahuje stejnou příponu, skládá se z čísla vzorku, časového razítka a z jednotlivých kanálů. Data nejsou od sebe nikterak oddělena a ani jednotlivé vzorky nejsou od sebe odděleny (např. u ASCII formátu jsou odděleny odřádkováním, každý vzorek = jeden řádek).

Veškerá data uložená v binárním datovém souboru jsou reprezentovaná v hexadecimálním formátu. Když je ukládáno dvou bytové slovo, tak se nejprve uloží méně významný bit (LSB) a potom více významný bit (MSB). Pro příklad bude využito dekadické číslo 1234, které bude uloženo ve formátu 3412. Pokud bude ukládáno čtyř bytové číslo, pak dekadické číslo 12345678 bude uloženo ve formátu 78563412.

Sekvence dat reprezentující binární datový soubor je následující:

- Číslo vzorku a časové razítko jsou ukládány v neznaménkovém binárním formátu po čtyřech bytech každý.
- Vzorek analogového kanálu je uložen do dvou binárních formátů, každý o 2 bytech. Nulová hodnota je uložena jako 0000 hexadecimálně, -1 je zaznamenána jako FFFF. Maximální kladná hodnota je 7FFFF a maximální záporná hodnota je 8001. Pro označení chybějících dat slouží 8000 hexadecimálně.
- Vzorek digitálního kanálu je ukládán ve skupinách

Délka souboru je různá, záleží na počtu kanálů a počtu vzorků v souboru. Počet bytů potřebných pro každý vzorek je následující: $(A_k \times 2) + (2 \times \text{INT}(D_m / 16)) + 4 + 4$,

kde A_k je počet analogových kanálů
 D_m je počet digitálních kanálů
 $\text{INT}(D_m/16)$ je počet digitálních kanálů dělených 16 a zaokrouhlených nahoru
4+4 4 byte pro číslo vzorku a 4 byte pro časové razítko

Zápis vzorku vypadá následovně: **n timestamp $A_1 A_2 \dots A_k S_1 S_2 \dots S_m$** ,

kde n je číslo vzorku, timestamp je časové razítko, A jsou analogové kanály a S jsou digitální kanály. Základní jednotkou časového razítka je μs (microseconds [mikrosekunda]).

Pokud bude v záznamu chybět časové razítko, pak bude tato pole nahrazeno hodnotou FFFFFFFF, z důvodu udržení celistvosti datového souboru.

Pro lepší pochopení zápisu hodnoty digitálního kanálu poslouží následující příklad:

Je 6 digitálních vstupů 0,0,0,0,1,1. Úkolem je tyto vstupy zapsat do binárního datového souboru.

Postup je následující:

- Zapsat digitální vstupy jako binární číslo (110000), které je vyjádřeno od nejméně významného bitu po nejvíce vážený.
- Doplnit hodnotu na 16 bitové číslo (0000 0000 0011 0000)
- Převést na hexadecimální hodnotu (00 30)
- Uložit data v podobě LSB/MSB formátu (30 00).

Malý příklad binárního souboru:

01 00 00 00 00 00 00 00 1E FC B5 04 64 00 1D 00 79 FF 3B FF
00 00 **02 00 00 00** A7 00 00 00 51 FC CF 04 5E 00 25 00 77 FF
ED FE 00 00 **03 00 00 00** 4E 01 00 00 8A FC E3 04 57 00 2D 00
75 FF A1 FE 20 00 **04 00 00 00** F5 01 00 00 C6 FC F1 04 50 00
34 00 74 FF 56 FE 10 00 **05 00 00 00** 9C 02 00 00 08 FD FA 04
48 00 3D 00 74 FF 0A FE 30 00 **06 00 00 00** 43 03 00 00 4F FD
FF 04 40 00 44 00 74 FF BF FD 00 00...

... **73 0C 00 00** 38 3E 00 00 8A 01 42 FE FF FF 00 00 FF FF
A7 FE 00 00 **74 03 00 00** DF 3E 00 00 7A 01 5F FE FE FF 00 00
FF FF 92 FE 00 00 **75 03 00 00** 85 3F 00 00 68 01 7D FE FE FF
00 00 FF FF 7F FE 00 00

[3]

2.3.4 Informační soubor

Informační soubor je nepovinný doplňkový soubor. Zajišťuje výměnu informací týkající se záznamů COMTRADE formátu tak, že může umožnit lepší analýzu zaznamenaných dat. Další informace jsou uloženy v odděleném souboru, který poskytuje plnou zpětnou kompatibilitu mezi současným a budoucím programem využívajícím soubory typu COMTRADE. Některé programy čtou data z informačních souborů proto, aby mohly lépe popsat některé veřejné části hlavičky, vstupů, nebo jiných dat, které jsou začleněny v tomto standardu, a mohou učinit nějaký zásah jako odezvu na čtená data.

Informační soubor je ASCII textový soubor a je to počítačově čitelný specifický formát. Soubor obsahuje informace, které jsou přístupné pro čtení běžným uživatelům, a informace, které jsou určeny pro čtení pouze vybrané skupině uživatelů. Tyto dvě skupiny informací jsou rozděleny jako veřejná a soukromá část, jednotlivě ležící v oddělených částech souboru. Data uložená v informačním souboru mohou být uložena v soukromé sekci, pokud je vhodně definovaná. Pokud vhodně předdefinovaná veřejná sekce není dostupná, pak mohou být data uložena v sekci soukromé.

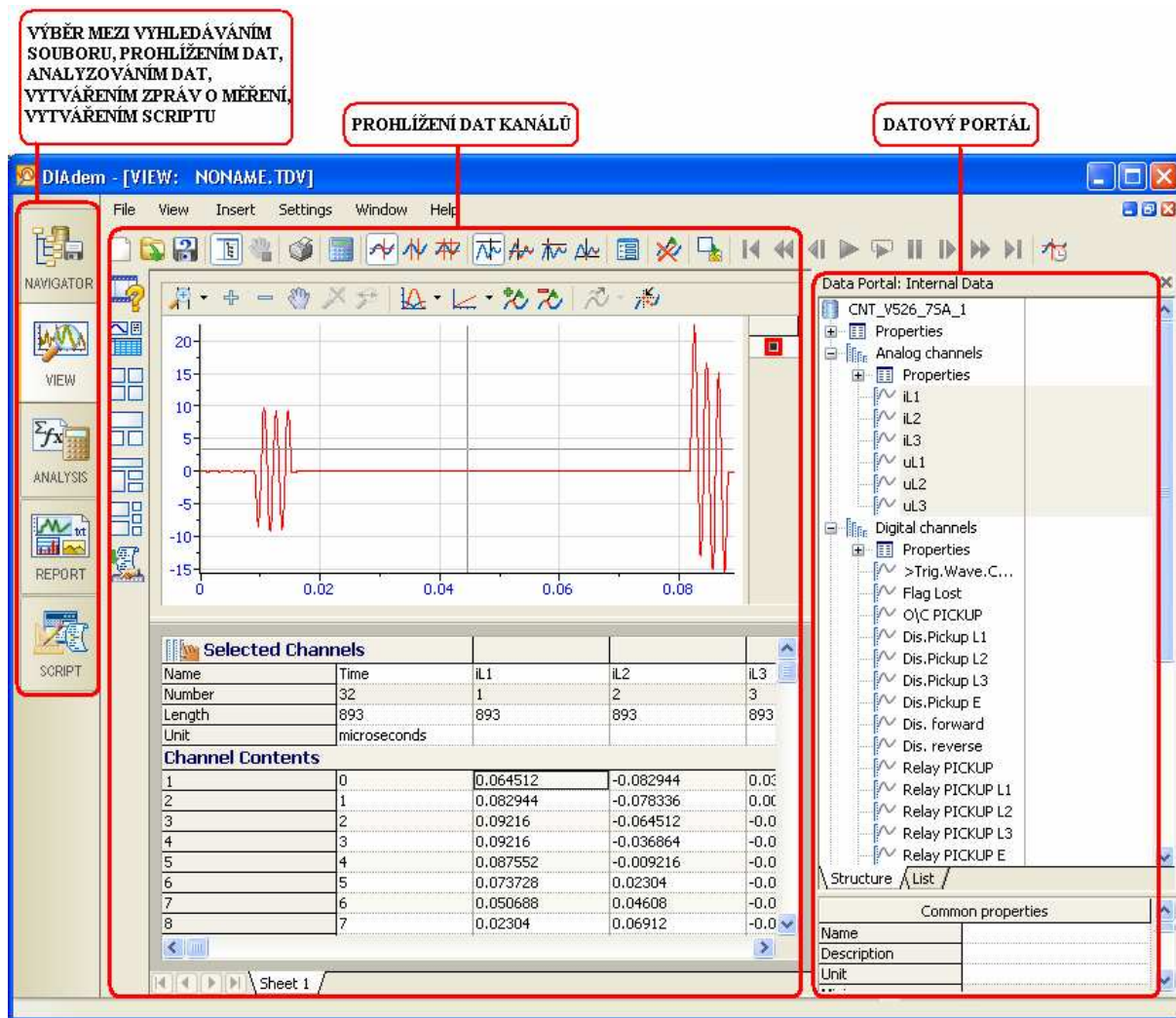
[3]

Struktura informačního souboru je obsažena v příloze II.

3 DIAdem

NI DIAdem je software, který je určen pro správu, analýzu a vykazování dat získaných při sběru dat, nebo dat vytvořených během simulace. DIAdem je navržen tak, aby odpovídal požadavkům, jak má vypadat dnešní testovací prostředí, pro které je charakteristický rychlý přístup k velkému množství zaznamenaných dat.

DIAdem nabízí matematicky a graficky interaktivní analyzování dat a reprezentaci dat v podobě zpráv. DIAdem poskytuje jedinečné kombinace nástrojů, které jsou jakoby ušité na míru potřebám techniky, vědy apod.



Obr. 4 Pracovní plocha modulu VIEW SW DIAdem

3.1 DIAdem DataFinder

Pro data uložená ve formátu TDM existuje prostředek k jejich indexaci. Tato indexace umožní jejich rychlé prohledávání. V tomto prostředku jsou implementovány obdobné technologie jako ve vyhledávacích službách na Internetu, které umožňují i velmi pokročilé prohledávání naměřených dat.

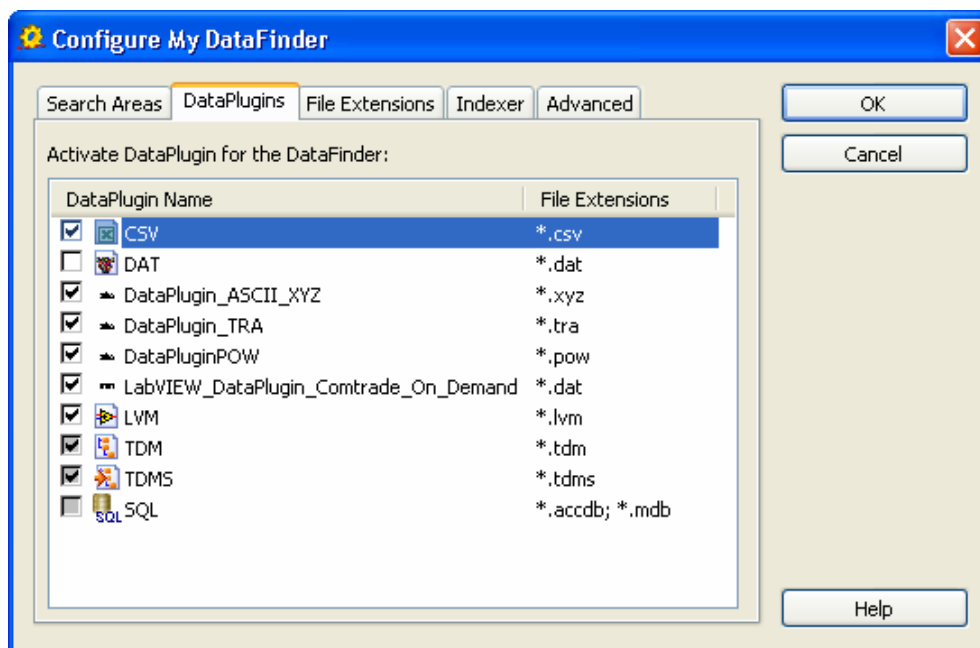
Po nainstalování DIAdem DataFinder proběhne na pozadí indexace všech datových souborů obsažených na disku počítače v úrovních odpovídajících standardu TDM:

- soubor
- skupina kanálů
- kanál

K této indexaci se využívá doby nečinnosti počítače v jeho zapnutém stavu. Ikonku umožňující prohlídku aktuálního stavu indexace najde uživatel na spodní liště OS Windows.

Konfigurace této služby je dostupná přes ikonu DataFinder pravým tlačítkem myši výběrem možnosti: *Properties*.

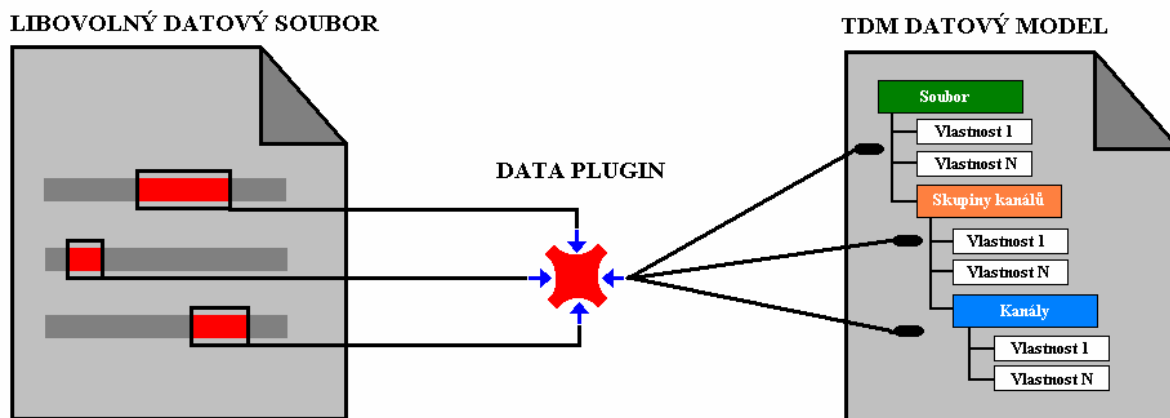
[4]



Obr. 5 Seznam DataPluginů zobrazených prostředkem DIAdem DataFinder

4 DataPlugin

Je to prostředek, který poskytuje jednotný způsob pro přístup k datům bez ohledu na datový formát. Na stránkách NI je ke stažení přes 70 DataPluginů. Výhodou je nejen to, že sjednocuje přístup k datovým souborům v LabVIEW, ale také to, že na základě mapování dat, založeného na TDM datovém modelu, otevírá všechny soubory až po dovolání dat pomocí DIAdem DataFinder. DataPluginy jsou programy, které definují formát souborů a dávají jim příponu.



Obr. 6 Mapování neznámého formátu s využitím TDM datového modelu

4.1 Typy DataPluginu

Zpočátku mohli být DataPluginy vytvářeny pouze pomocí programovacích jazyků jako C++ nebo VBScript. National Instrument proto vytvořil LabVIEW DataPlugin SDK, který umožňuje vytvářet DataPluginy prostřednictvím LabVIEW G-kódu.

V LabVIEW je možné vytvářet dva typy DataPluginů:

- **One Shot DataPlugins** – vyznačuje se svou jednodušší formou, v programu je pouze jedno VI, které je upravováno programátorem dle struktury daného datového formátu. One Shot DataPlugin zpracovává všechny popisné informace stejně tak dobře jako datové hodnoty všech kanálů. Programátorem modifikované VI je použito v programu pouze jednou, v čase kdy se načítají všechna data prostřednictvím DataPluginu.
- **On Demand DataPlugins** – jedná se o více komplikovaný a náročnější způsob pro vytváření DataPluginu. On Demand DataPlugin obsahuje dvě volání, *yourCodeHere_meta.vi*, který je volán pouze jednou během provádění DataPluginu a pak *yourCodeHere_raw.vi*, který může být volán jednou, ale i vícekrát. Záleží to na tom, kdy jsou vyžadována data. *YourCodeHere_meta.vi* deklaruje datové kanály, které jsou specifické například datovým typem, názvem, velikostí (délkou), popisnými vlastnostmi, ale neobsahují žádné data. *YourCodeHere_raw.vi* je volán individuálně pro každý datový kanál.

Porovnání One Shot a On Demand:

Výhodou One Shot DataPluginu je to, že časté načítání všech hodnot všech kanálů je rychlejší než v případě On Demand DataPluginu. One Shot DataPlugin je dobrou volbou, pokud je načítáno malé množství dat ze souboru. Samotné načtení všech hodnot ze všech kanálů pak zabere jen několik sekund. Pokud je však známo, že soubor obsahuje velké množství dat, pak je výhodnější použít On Demand DataPlugin, protože ten nemusí načítat vždy všechna data ze souboru, ale umožňuje načíst ty data, kterou jsou momentálně požadována. Pokud kanál obsahuje 100 000 záznamů, ale pro analýzu stačí jen 200 záznamů z tohoto kanálu, pak je možné vyčíst jen 200 záznamů. Není potřeba vyčíst všechna data z kanálu. U One Shot DataPluginu tomu tak není, protože ten kdykoliv je volán vyčítá veškerá data všech kanálů, kterou jsou v souboru uloženy.

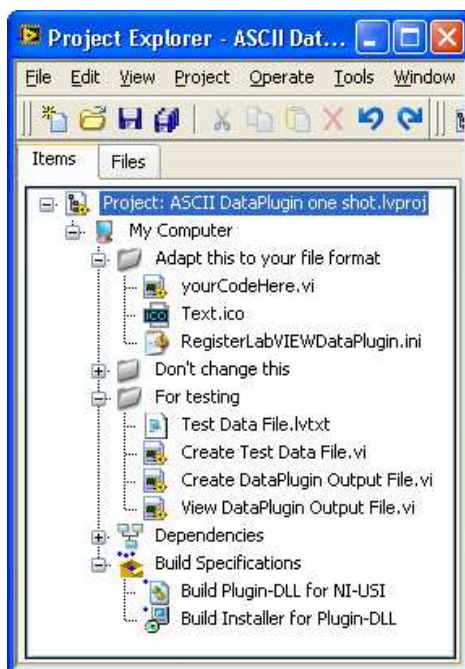
Při samotném výběru typu DataPluginu je nutné si uvědomit, pro jaké soubory bude tento DataPlugin využíván, zda se bude jednat o soubory, které obsahují malé množství dat, nebo o soubory, které jsou velmi rozsáhlé. Dále to může taky ovlivňovat skutečnost, zda záleží na době, během které se data načtou do RAM paměti.

[5]

4.2 Vytváření DataPluginu

Před samotným vytvářením DataPluginu je nutné nainstalovat LabVIEW DataPlugin SDK, který je ke stažení na <http://zone.ni.com/devzone/cda/epd/p/id/6016#0requirements>. Zde je na výběr ze dvou instalátorů, jeden je určen pro verzi LabVIEW 8.6, druhý pro verzi LabVIEW 2009.

DataPlugin v LabVIEW se vytváří prostřednictvím LabVIEW projektu. Nejlepším způsobem je využít nějaký dřívější DataPlugin, ve kterém se navrhne nové jádro specifického formátu datového souboru. Znázornění struktury celého projektu je viditelné na obr. 7.



Obr. 7 DataPlugin projekt

Jak lze vidět z obr. 7, DataPlugin se skládá z několika adresářů, ve kterých se nacházejí navržené VI a další soubory nezbytné pro vývoj DataPluginu. Jednotlivé adresáře budou následně popsány:

- *Adapt this to your file format* : obsahuje zdrojový kód pro čtení dat z konkrétního datového souboru, který má tento vytvářený DataPlugin podporovat. V případě one-shot to je *yourCodeHere.vi*, v případě on-demand se jedná o soubory *yourCodeHere.meta.vi* a *yourCodeHere.raw.vi*.

Soubor *RegisterLabVIEWDataPlugin.ini* se používá pro editaci přiřazené přípony souboru, názvu DataPluginu a názvu knihovny.

- *Don't Changes this* : obsahuje několik souborů, které tvoří základ celé architektury DataPluginu a nesmí být nikdy upravovány ani vymazány z projektu.
- *For Testing* : obsahuje několik souborů, které mohou být užitečné při ladění a testování DataPluginu.
- *Dependencies* : obsahuje nezbytně nutné soubory pro vytvoření knihovny a instalátoru.
- *Build Specifications* : obsahuje konfiguraci pro vytvoření knihovny DataPluginu DLL a pro vytvoření instalátoru DataPluginu. K sestavení knihovny je potřeba všech souborů z projektu a samotná knihovna, ještě s několika soubory, slouží pro sestavení instalátoru DataPluginu, podporujícího danou příponu datového souboru.

[5]

4.3 Testování a ladění DataPluginu

V adresáři *Testing* je několik souborů, z nichž některé umožňují otestovat vytvořený DataPlugin a tak ověřit, zda je vytvořený zdrojový kód plně funkční a tudíž podporuje veškeré datové soubory s odpovídající příponou.

Prvním takovým souborem je *CreateDataPluginOutputFile.vi*. Toto VI simuluje architekturu DataPluginu volající *yourCodeHere.vi*, které bylo programátorem upraveno, a poté zobrazuje nahrání výsledných dat v pop-up, interaktivním VI. Toto VI je pak možno použít k ladění kódu.

Druhým souborem je *ViewDataPluginOutputFile.vi*, které je ve skutečnosti pop-up interaktivní VI zobrazující data nahraná prostřednictvím DataPluginu v simulačním módu.

[5]

5 DataPlugin one shot

5.1 DataPlugin proprietárního datového formátu

Tento DataPlugin byl vytvořen, aby na jednoduchém příkladu byl vysvětlen přístup k vývoji DataPluginu. DataPlugin bude podporovat doposud nepodporovaný souborový formát s příponou *.xyz.

5.1.1 Struktura souboru data.xyz:

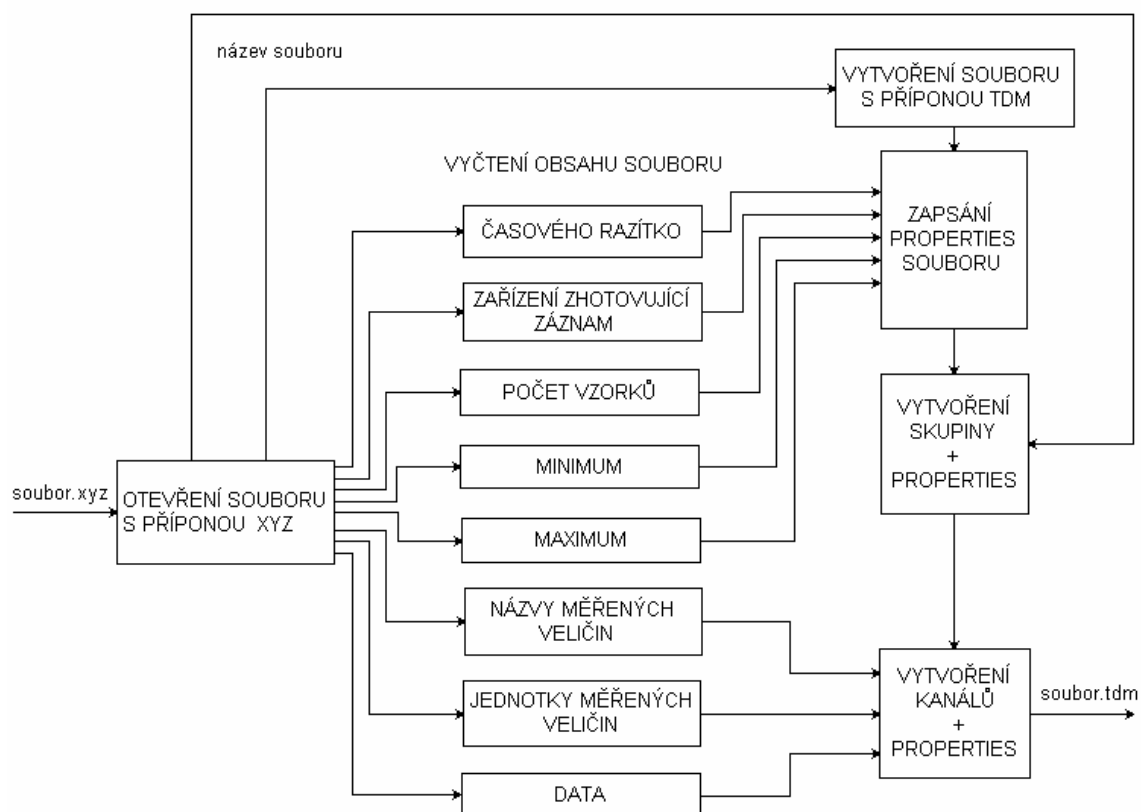
Struktura tohoto souboru je níže znázorněna na konkrétních zaznamenaných datech. Jedná se o soubor, který obsahuje určitou hlavičku, která je složena z časového razítka, názvu zařízení zhotovujícího záznam, z celkového počtu vzorků, minima, maxima, z názvu jednotlivých kanálů a jejich jednotek. Po hlavičce pak již následují pouze hodnoty jednotlivých kanálů.

18:30:00 30.11.2009			
excel 2003			
720	-1	1	
Time	Uhel	Sin	Cos
s	°	bezrozměrů	bezrozměrů
11/30/2009 18:30:00.0001	1	0,017452	0,999848
↓	↓	↓	↓
11/30/2009 18:30:00.0720	720	0,000000	1,000000

5.1.2 Návrh jádra DataPluginu

K vytvoření DataPluginu byl editován starší projekt typu one shot, což je jedna z možností tvorby DataPluginu. Struktura takového projektu je popsána v kapitole 4.2.

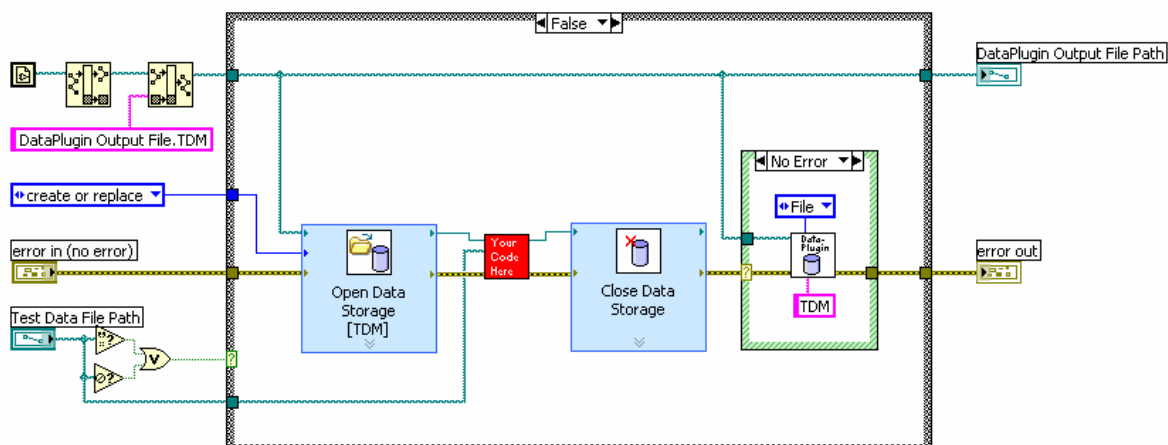
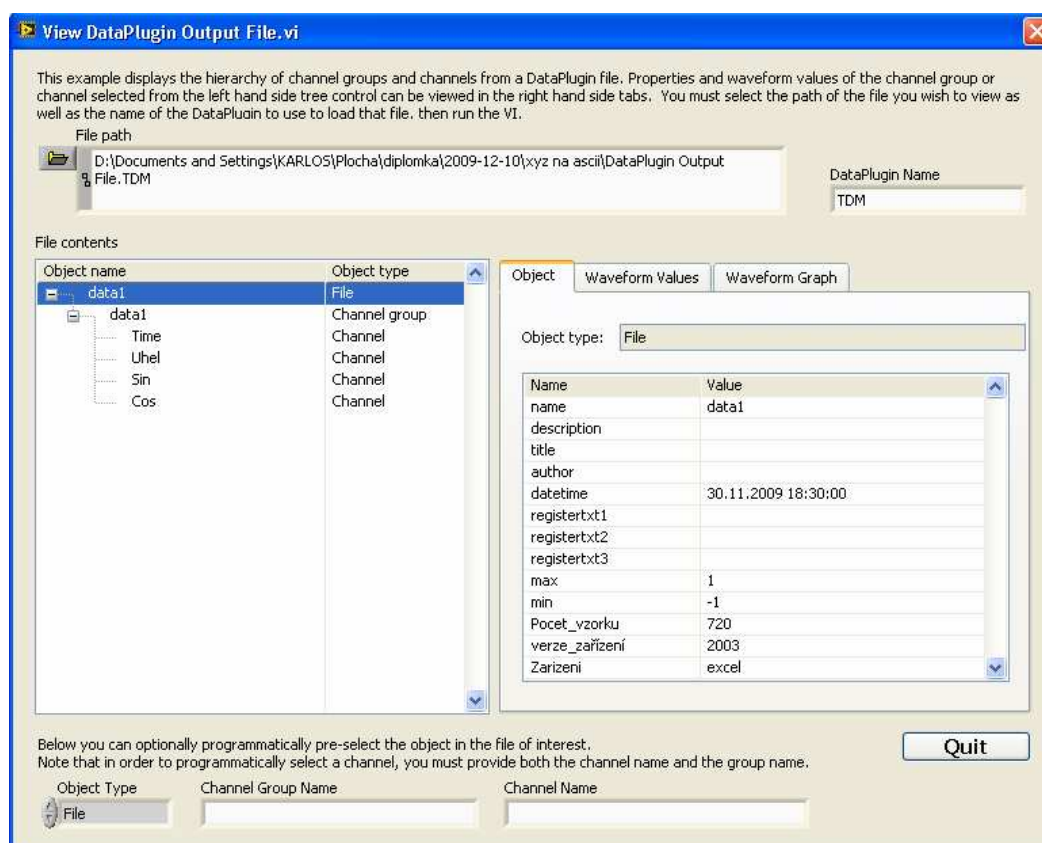
Pro vytvoření DataPluginu, umožňujícího v DIAdemu podporu souborového formátu *.xyz, je potřeba navrhnout jádro DataPluginu. Tím jádrem je *yourCodeHere.vi*. Toto VI je tvořeno schématem, které otevře doposud nepodporovaný datový soubor a vyčte z něj data. Vyčtená data pak zapíše do již podporovaného formátu (např. TDM). K tomu, aby byly data po otevření nepodporovaného formátu dobře vyčteny, je důležité, aby programátor dobře znal strukturu tohoto formátu. Data jsou vyčítána ze souboru po jednotlivých řádcích do datového pole typu string. V řádku se nachází zpravidla více údajů, které je potřeba správně rozpoznat. V tomto případě se jedná o ASCII soubor. Jednotlivá data v řádku jsou mezi sebou oddělena tabulátorem. Poté, co všechna data jsou správně vyčtena ze souboru, jsou tyto data zapsána do struktury souborového formátu *.TDM. Toto je podstata tohoto VI, jehož návrh je na přiloženém CD.



Obr. 8 Blokový diagram jádra DataPluginu

5.1.3 Testování

Pro testování byl použit jako testovací soubor data1.xyz. K otestování funkčnosti vytvořeného DataPluginu se využívá *CreateDataPluginOutputFile.vi* (ten je na obr. 9), ve kterém se zadá cesta k testovanému souboru. Po spuštění tohoto VI dojde k otestování navrženého kódu v *yourCodeHere.vi* a otevře se viewer (prohlížeč dat), který je zobrazen na obr. 10, a který zobrazuje obsah testovaného souboru.

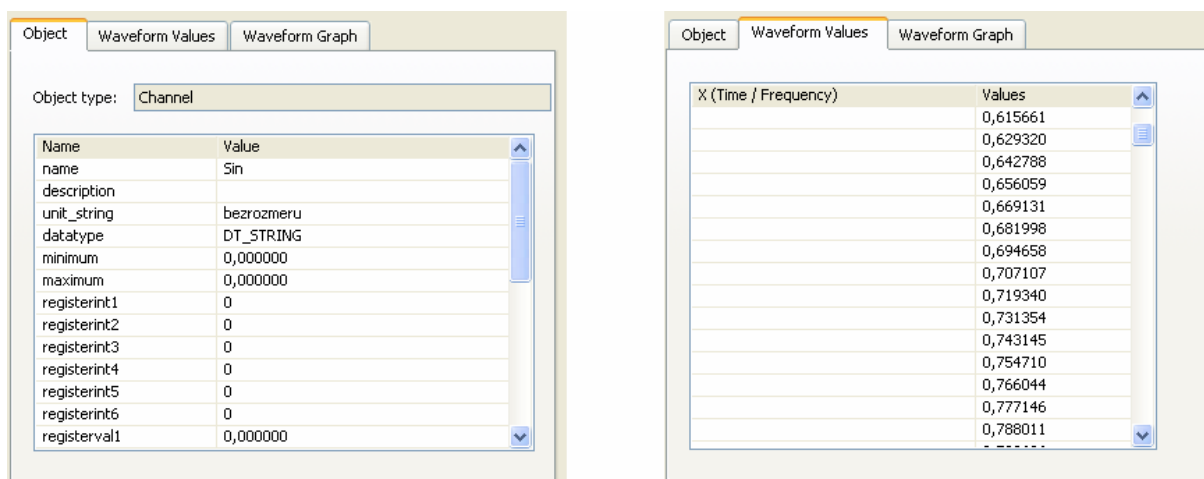
**Obr. 9** CreateDataPluginOutputFile.vi

Obr. 10 Prohlížeč dat ze souboru data1.xyz po spuštění DataPluginu

Viewer je nejjednodušší cestou, jak ověřit funkčnost DataPluginu. Viewer se skládá z několika částí. Na levé straně je znázorněn obsah souboru, který je tvořen z následujících údajů.

- Název souboru (*File...data1*)
- Název skupiny kanálů (*Channel group...data1*)
- Názvy samotných kanálů (*Channel...Time, Uhel, Sin, Cos*)

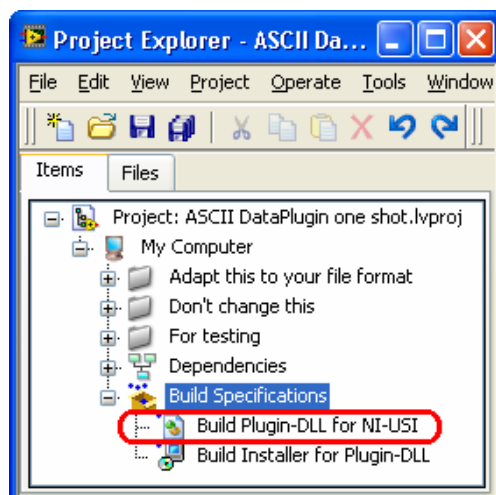
Jedná se v podstatě o strukturu založenou na principu TDM datového modelu, na kterém je položen základ DIAdemu. Pravá strana vieweru je oblast, ve které se zobrazují properties (vlastnosti) souboru, skupin a jednotlivých kanálů, kterými mohou být například jednotky daných veličin, minimum, maximum, časové razítko, název zařízení apod. Properties souboru jsou viditelné po pravé straně na obr. 10, properties a hodnoty kanálu jsou na obr. 11.



5.1.4 Vytváření knihovny DataPluginu

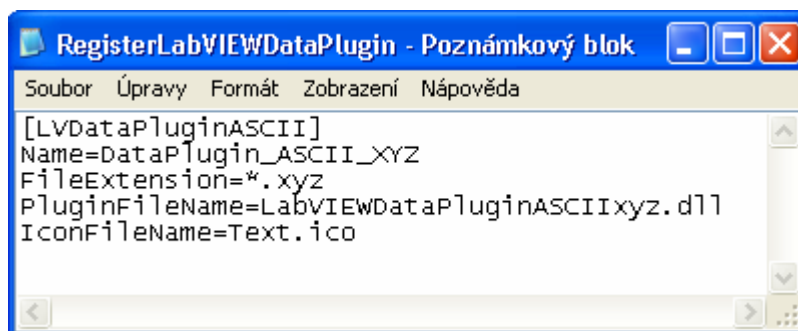
Pokud testování DataPluginu bylo úspěšné, tzn. pokud DataPlugin funguje správně, pak je namístě vytvořit knihovnu a instalátor podpory datového formátu tak, aby mohla být tato podpora doinstalována do systému a následně plně využívána v softwaru DIAdem.

Pro vytvoření knihovny se využívá volby ve struktuře projektu DataPluginu, zobrazené na obr. 12 v červeném rámečku.



Obr. 12 Vytváření knihovny

Před samotným vytvořením knihovny je ještě podstatné editovat *RegisterLabVIEWDataPlugin.ini* v adresáři *Adapt this to your file format..* Edituje se pouze několik málo údajů jako je např. jméno DataPluginu, souborová přípona, pro kterou je určen, název knihovny pro příslušný DataPlugin, viz. obr. 13.



Obr. 13 Konfigurace *RegisterLabVIEWDataPlugin.ini*

Po dokončení editování souboru *RegisterLabVIEWDataPlugin.ini* je dalším krokem otevřít průvodce pro tvorbu knihovny, tj. dvojím pokliknutím na *Build Plugin-DLL for NI-UI*. Po otevření okna bude nezbytné nastavit několik vlastností. Nejdůležitější vlastnosti, které je třeba zadat, jsou zachyceny na obr.14 a obr.15.

Obr. 14 Nastavení názvu DataPluginu a cílového adresáře

V záložce *Information* se vyplní konečný název knihovny, která bude vytvořena a cesta do adresáře, ve kterém bude uložena. V záložce *Version information* se dle potřeby upraví číslo verze.

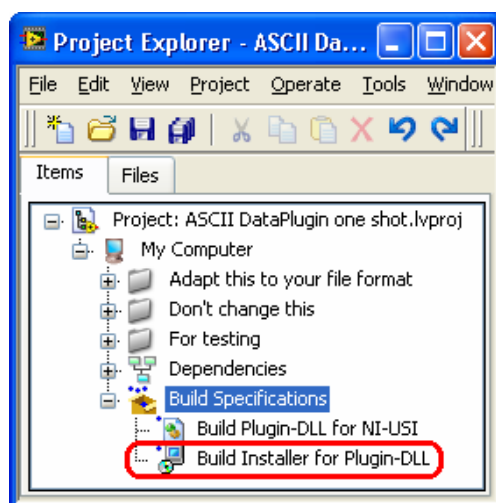
Obr. 15 Version information

Knihovna se vytvoří po stisku tlačítka *Build*.

Obr. 16 Tlačítko *Build*

5.1.5 Vytváření instalátoru DataPluginu

Pro vytvoření instalátoru se využívá volby ve struktuře projektu DataPluginu, zobrazené na obr. 17 v červeném rámečku.

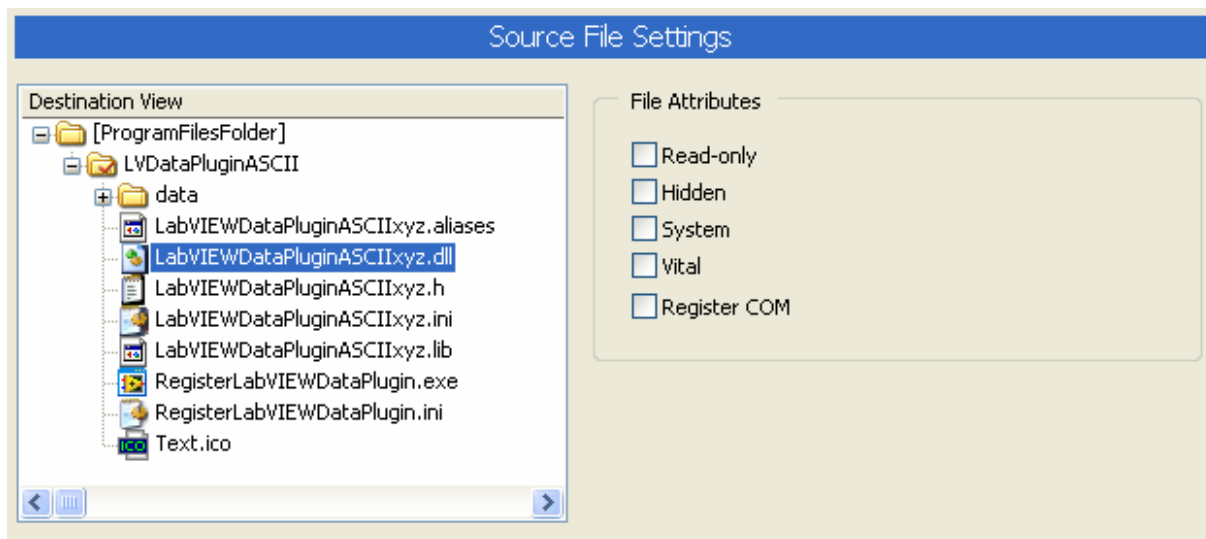


Obr. 17 Vytváření instalátoru

V záložce *Product Information*, zobrazené na obr. 18, se provádí nastavení názvu instalátoru, cílového adresáře a čísla verze produktu (číslo by mělo být shodné s číslem verze při vytváření knihovny).

Obr. 18 Nastavení názvu DataPluginu a cílového adresáře

V záložce *Source File Settings*, zobrazené na obr. 19, dochází k nastavení zdrojového souboru, jakožto je např. knihovna (vytvořením knihovny se zabývá předešlá kapitola).



Obr. 19 Nastavení zdrojového souboru

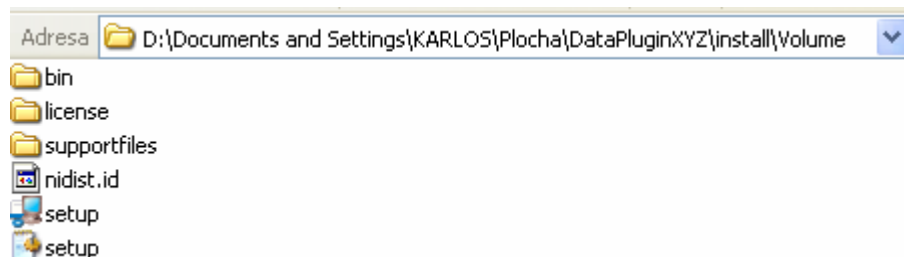
Pokud jsou všechny potřebné parametry řádně nastaveny a vše je připraveno pro sestavení instalátoru, pak stačí pouhým stisknutím tlačítka *Build* zahájit vytvoření instalátoru.



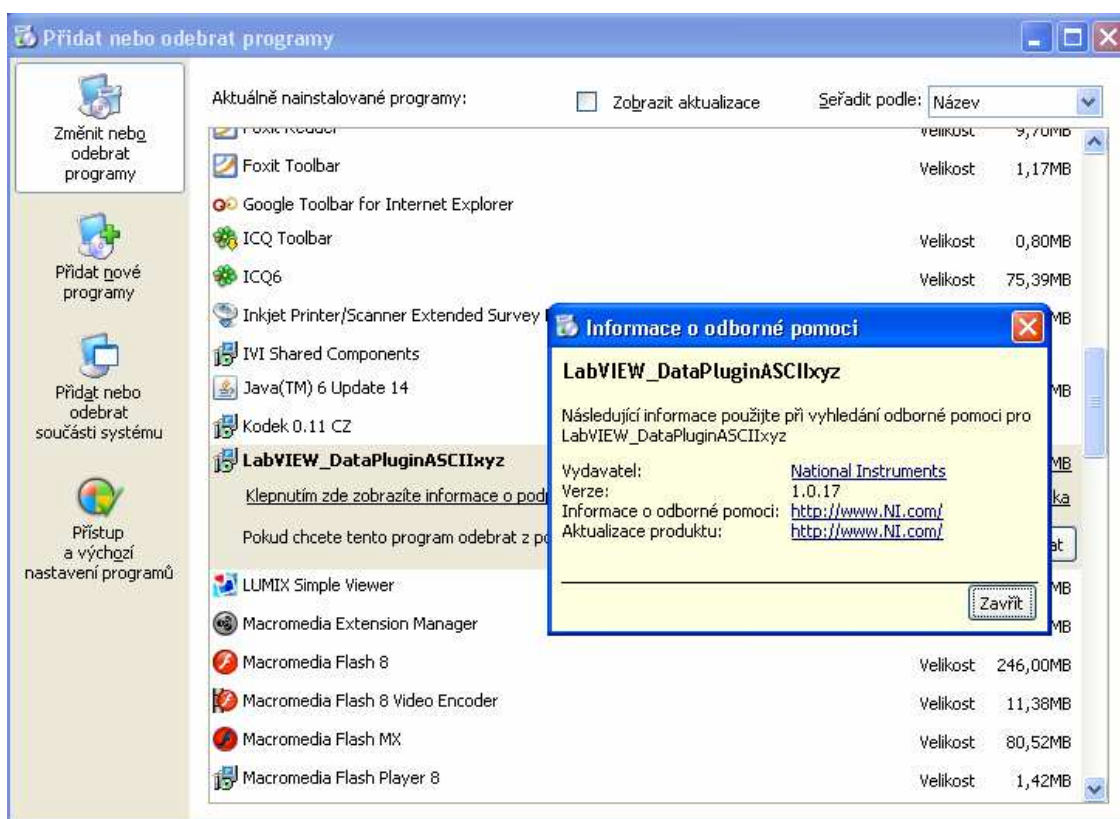
Obr. 20 Tlačítko *Build*

5.1.6 Instalace DataPluginu do systému

V adresáři, do kterého byl uložen vytvořený instalátor *DataPluginASCIIxyz*, najdeme instalační soubory (obr. 21). Souborem *setup.exe* se zavádí instalace podpory datového formátu *.xyz. Po nainstalování DataPluginu je možné v panelu *Přidat/Odebrat programy* ověřit nainstalování softwarové podpory pro daný formát souboru (obr. 22). Pokud by, již po nainstalování softwarové podpory, nastala situace, že je potřeba ještě nějaké editace DataPluginu, je nutné ze seznamu nainstalovaných programů a aplikací DataPlugin odinstalovat, provést editaci, znovu vytvořit jak knihovnu tak instalátor, a následně znovu nainstalovat již upravený DataPlugin.



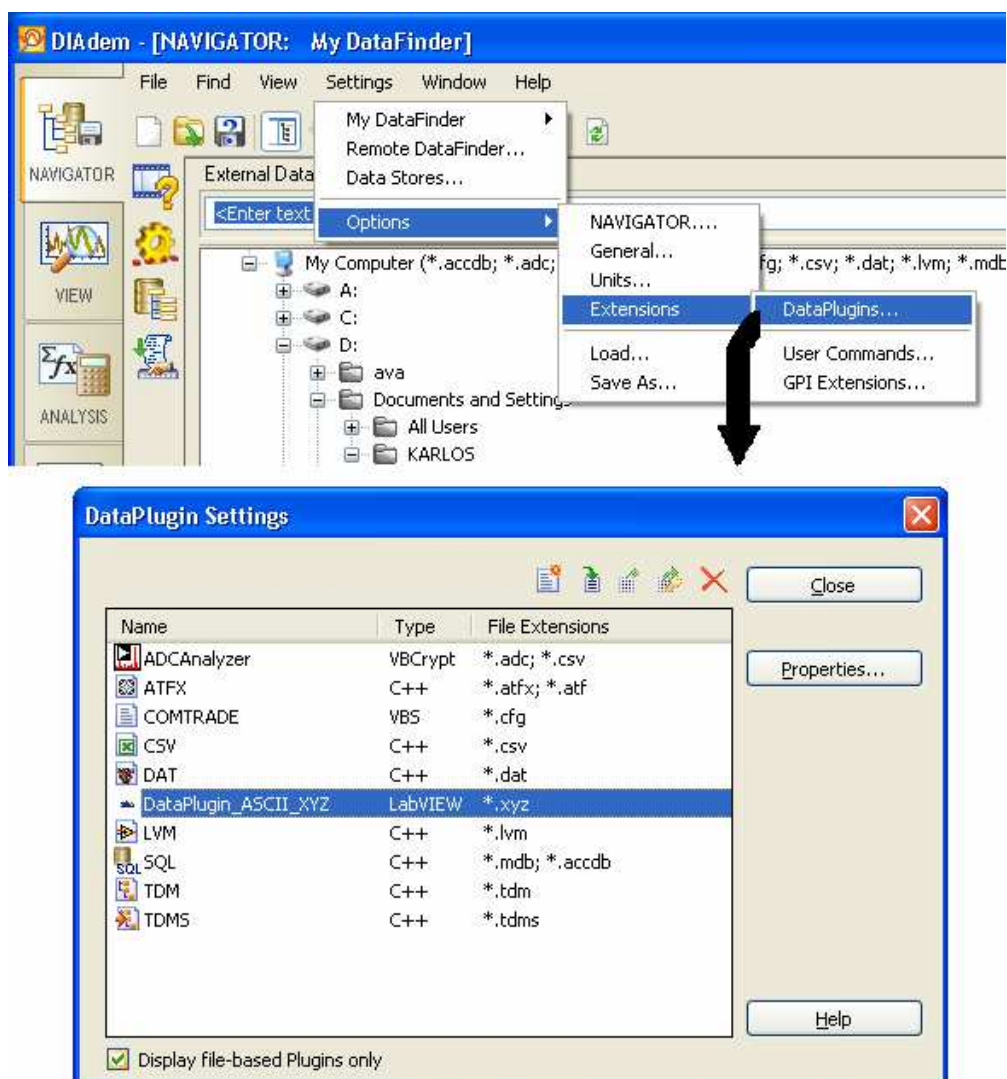
Obr. 21 Obsah instalačního adresáře DataPluginu



Obr. 22 Instalace DataPluginu *DataPluginASCIIxyz*

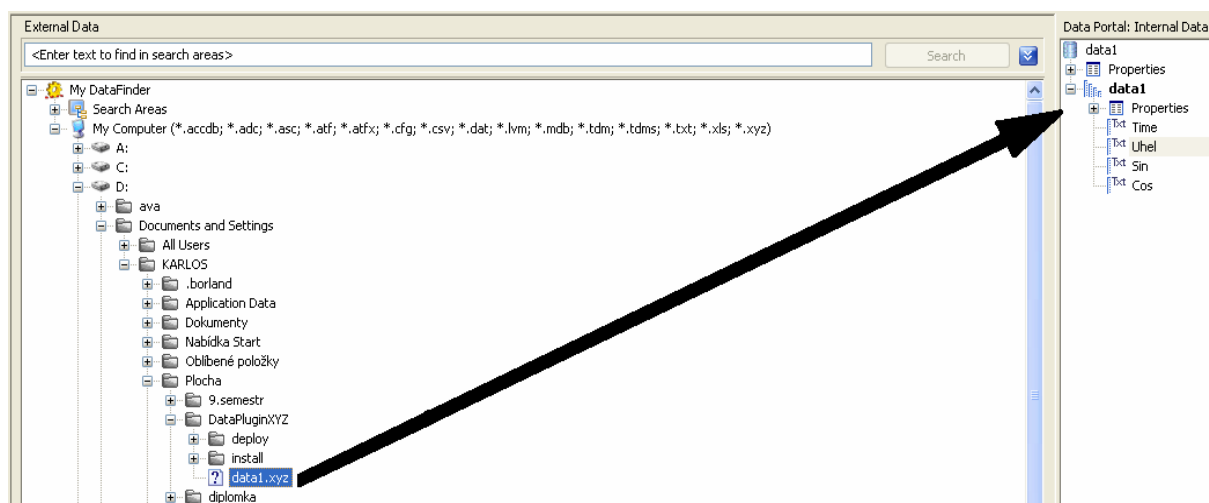
5.1.7 Použití DataPluginu v DIAdemu

Po spuštění DIAdemu je možné ověřit, jestli je v seznamu podporovaných datových formátů začleněn i programátorem vytvořený DataPlugin podporující soubory s příponou xyz. Postup pro otevření seznamu DataPluginů je na obr. 23. Jak lze na obrázku dobře vidět, DataPlugin podporující xyz je v seznamu, což znamená, že v DIAdemu je možné pracovat se všemi datovými soubory s příponou xyz.



Obr. 23 Podpora souborového formátu *.xyz v DIAdemu

Pro načtení souboru v DIAdemu je potřeba nejprve daný soubor nalézt pomocí DataFinderu a pak jej myší přetáhnout do datového portálu (*Data portal*), nacházejícího se v pravé části v záložce *Navigator* (viz. obr. 24).



Obr. 24 Načtení souboru data1.xyz

Data obsažená v souboru data1.xyz je možné vidět na obr. 25 rozdělené do kanálů. Podle údajů, které jsou zobrazeny v kanálech a properties (vlastnostech), se dá říci, že podpora datového formátu je plně funkční.

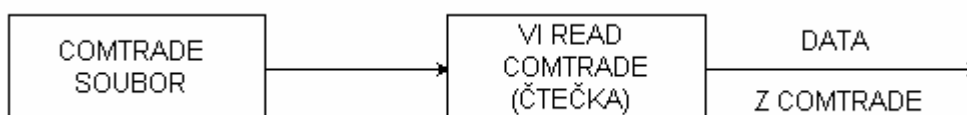
<All Channels>				
Name	Time	Uhel	Sin	Cos
Number	1	2	3	4
Length	720	720	720	720
Unit	s	°	bezrozměrů	bezrozměrů
Channel Contents				
1	11/30/2009 18:30:00.0001	1	0,017452	0,999848
2	11/30/2009 18:30:00.0002	2	0,034899	0,999391
3	11/30/2009 18:30:00.0003	3	0,052336	0,998630
4	11/30/2009 18:30:00.0004	4	0,069756	0,997564
5	11/30/2009 18:30:00.0005	5	0,087156	0,996195
6	11/30/2009 18:30:00.0006	6	0,104528	0,994522
7	11/30/2009 18:30:00.0007	7	0,121869	0,992546
8	11/30/2009 18:30:00.0008	8	0,139173	0,990268
9	11/30/2009 18:30:00.0009	9	0,156434	0,987688
10	11/30/2009 18:30:00.0010	10	0,173648	0,984808
11	11/30/2009 18:30:00.0011	11	0,190809	0,981627
12	11/30/2009 18:30:00.0012	12	0,207912	0,978148
13	11/30/2009 18:30:00.0013	13	0,224951	0,974370
14	11/30/2009 18:30:00.0014	14	0,241922	0,970296
15	11/30/2009 18:30:00.0015	15	0,258819	0,965926
16	11/30/2009 18:30:00.0016	16	0,275637	0,961262
17	11/30/2009 18:30:00.0017	17	0,292372	0,956305
18	11/30/2009 18:30:00.0018	18	0,309017	0,951057
19	11/30/2009 18:30:00.0019	19	0,325568	0,945519
20	11/30/2009 18:30:00.0020	20	0,342020	0,939693
21	11/30/2009 18:30:00.0021	21	0,358368	0,933580

Obr. 25 Obsah souboru data1.xyz rozdělený do jednotlivých kanálů

5.2 DataPlugin podporující souborový formát COMTRADE

Před vytvářením DataPluginu bylo nutné nejprve vytvořit VI pro zapsání dat ve formátu COMTRADE , a poté VI pro čtení těchto dat ze souboru ve formátu COMTRADE. V kapitole 5.2.1 bude popsán návrh pro zapisování dat a v kapitole 5.2.2 bude popsán návrh pro čtení dat.

Na obr. 26 je blokový diagram, který vysvětluje význam vytvořené čtečky dat formátu COMTRADE. Tato čtečka byla vytvořena za účelem vyčíst data ze struktury souboru formátu COMTRADE do datových polí. Datových polí je několik, jsou to pole pro analogová data, properties analogových kanálů, digitální data, properties digitálních kanálů, properties souboru a properties skupin kanálů. Tyto pole jsou dále využívány v rámci vytváření DataPluginu.



Obr. 26 Blokový diagram čtení dat z COMTRADE souboru

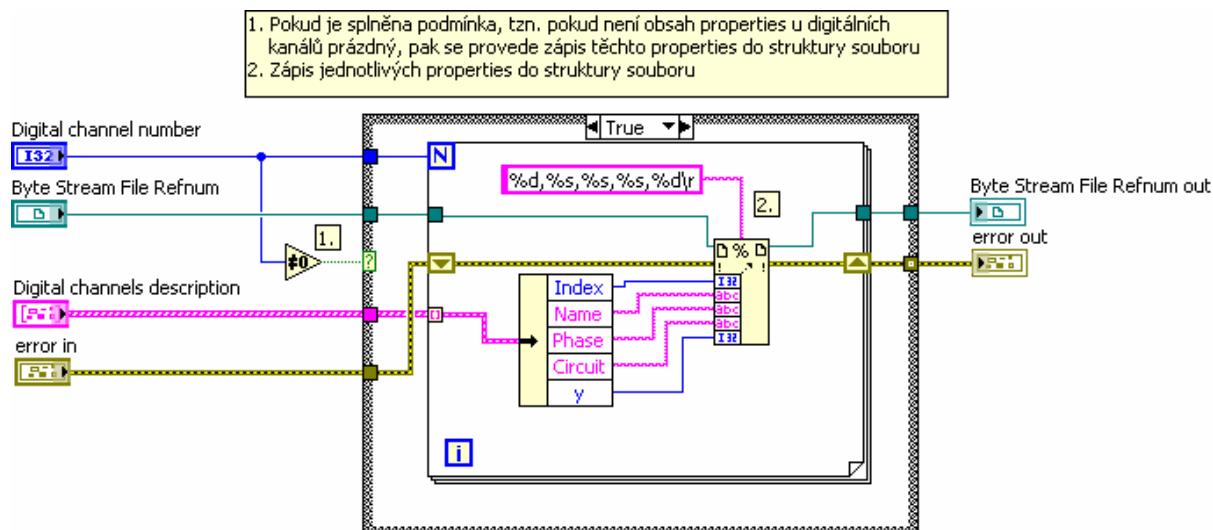
Oba návrhy, tj. návrh pro zápis i návrh pro čtení ze souboru datového formátu COMTRADE , jsou opatřeny error clustery, které slouží k zachycení vzniklé chyby v návrhu. V případě chyby (erroru) dojde k přerušení činnosti VI a k zobrazení identifikačního čísla erroru, který nastal.

5.2.1 Návrh VI pro zapisování dat ve formátu COMTRADE

K vytváření návrhu je potřeba dokonale znát strukturu souboru formátu COMTRADE. Nejde jen o to, co a na jakém řádku má být v souboru zapsáno, ale záleží i na datovém typu, minimu a maximu jednotlivých properties a také na způsobu oddělení dat v souboru. Celý návrh blokového schématu je rozdělen na dvě části. První částí je zapsání dat do konfiguračního souboru *.cfg, druhou částí je zapsání dat analogových a digitálních kanálů do datového souboru *.dat. Návrh je rozčleněn do několika subVI. Díky tomu se návrh stává přehlednější, je jednodušší v něm odhalit chyby a rychleji se v něm orientovat. Celý návrh je v příloze na CD.

SubVI bylo vytvořeno u následujících částí návrhu:

- Zápis properties analogových kanálů
- Zápis properties digitálních kanálů (viz. obr. 27)
- Zápis hodnot analogových kanálů
- Zápis hodnot digitálních kanálů
- Zápis dat do souboru jak v ASCII, tak i v binární podobě



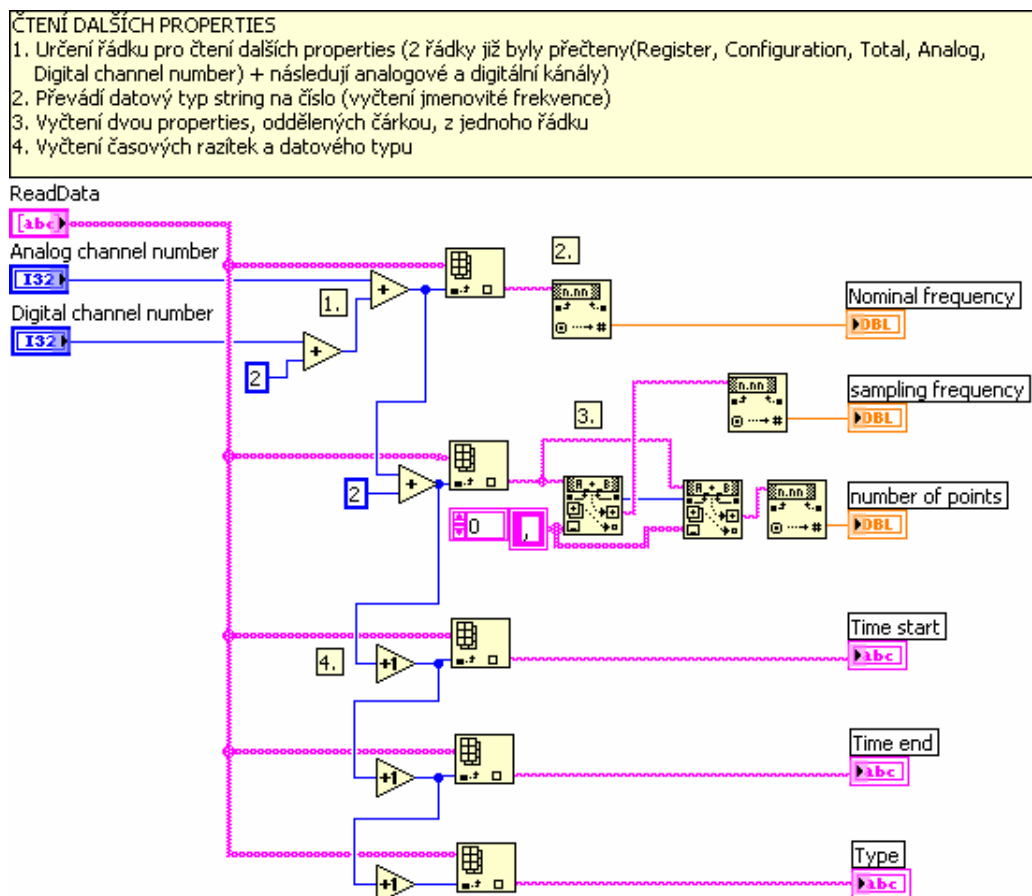
Obr. 27 Zápis properties (vlastností) digitálních kanálů

5.2.2 Návrh VI pro čtení dat ze souboru COMTRADE

Tak jako v kapitole 5.2.1 i zde je velmi důležitá dokonalá znalost formátu COMTRADE. Celý návrh je zde opět rozdělen na dvě části: první částí je čtení dat z konfiguračního souboru *.cfg, druhou částí je čtení dat z datového souboru *.dat. Oba soubory jsou otevřené pouze ke čtení, tudíž data v souboru se nikterak nemění. U čtení dat z konfiguračního souboru jsou čteny properties souboru, analogových a digitálních kanálů. U datového souboru záleží na způsobu zápisu dat v souboru (zda, se jedná o ASCII nebo binární soubor), a podle toho se využívají rozdílné části návrhu pro čtení dat z datového souboru.

Vytvořené subVI jsou následující:

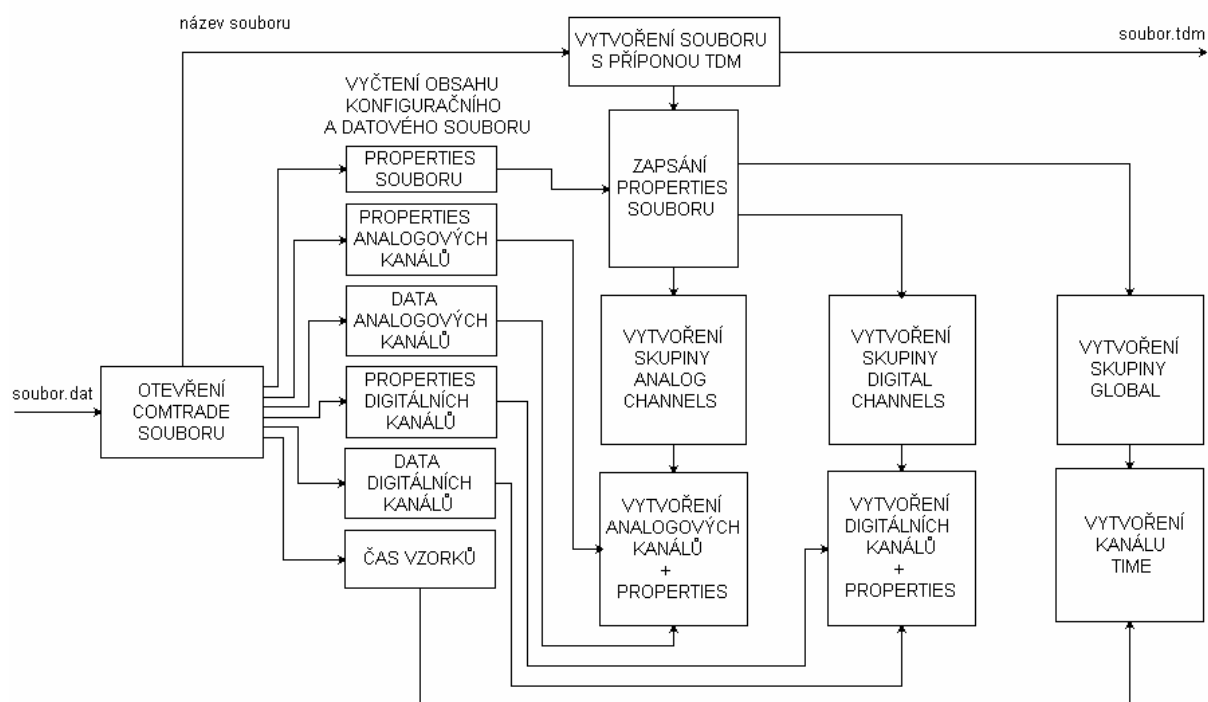
- Čtení properties zapsaných ve struktuře souboru (konkrétně se jedná o jmenovitou frekvenci, počet vzorků na periodu, vzorkovací frekvenci, časové razítko začátku a konce snímání vzorků, a typ souboru – ASCII nebo BINARY) – viz. obr. 28
- Čtení properties analogových kanálů
- Čtení properties digitálních kanálů
- Čtení hodnot analogových a digitálních kanálů ze souboru zapsaných v ASCII
- Čtení hodnot analogových kanálů ze souboru zapsaných v binární podobě
- Čtení hodnot digitálních kanálů ze souboru zapsaných v binární podobě



Obr. 28 Čtení properties souboru

5.2.3 Návrh jádra DataPluginu one shot pro podporu COMTRADE

Pro vytvoření jádra DataPluginu bylo využito VI pro čtení dat ze souboru, které je popsané v předchozí kapitole, tj. 5.2.2. Toto VI je pojmenováno jako *Read_Comtrade*. *Read_Comtrade* bylo upraveno jako subVI, aby se dalo vložit do jádra. Z tohoto VI byly v jádru vyčteny veškeré informace uložené v souborech *.cfg a *.dat. Tyto informace jsou nezbytné pro vytvoření DataPluginu. Vyčtené informace se dělí na informace týkající se pouze souboru (properties), informace skupin kanálů (properties), informace analogových kanálů (properties + hodnoty jednotlivých vzorků kanálů) a informace digitálních kanálů (properties + hodnoty jednotlivých vzorků kanálů). Tyto data byly zapsány do souboru TDM. Postup vytváření souboru TDM byl následující: Nejprve byl vytvořen TDM soubor stejného názvu, jako měl soubor ve formátu COMTRADE. Po zapsání properties souboru následovalo vytvoření skupin kanálů (analog a digital) a přiřazení jejich properties. V neposlední řadě bylo potřeba vytvořit analogové a digitální kanály, do kterých byly zapsány jak hodnoty tak i properties. Dále byla vytvořena skupina Global, do které byl zapsán kanál obsahující časové razítko (čas sejmутí vzorku v ms). Aby bylo možné data, které obsahuje kanál, zobrazovat v grafu, je potřeba nastavit všechny properties týkající se grafů. Mezi jednu z nejdůležitějších patří velikost inkrementu, což je velikost časového kroku mezi dvěma následujícími vzorky. Blokové schéma jádra DataPluginu je v příloze na CD.



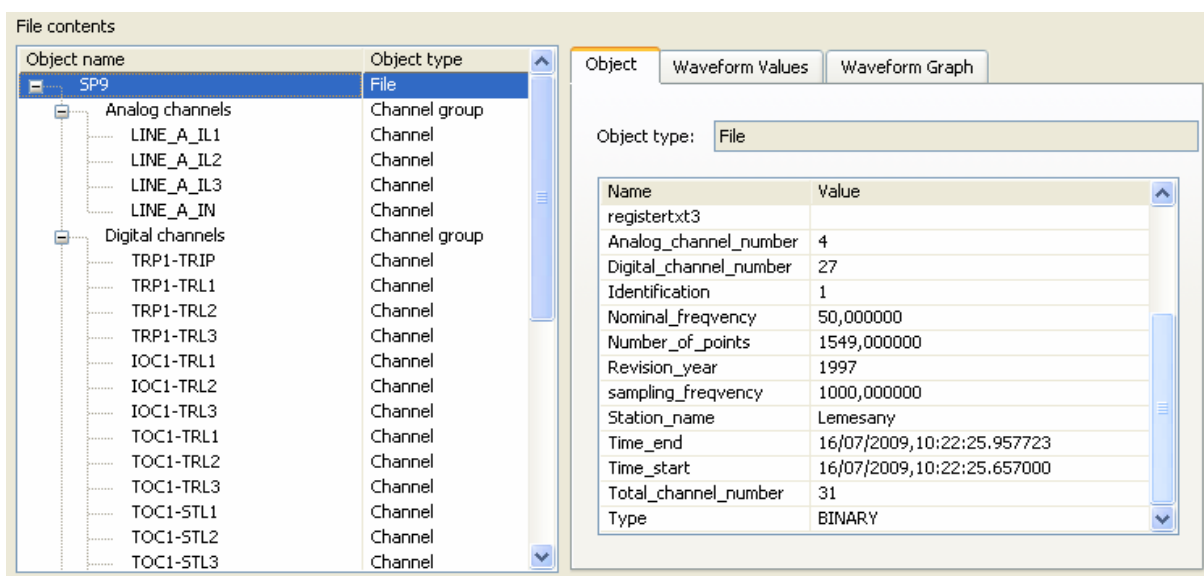
Obr. 29 Blokový diagram DataPluginu COMTRADE

5.2.4 Testování DataPluginu podporujícího COMTRADE

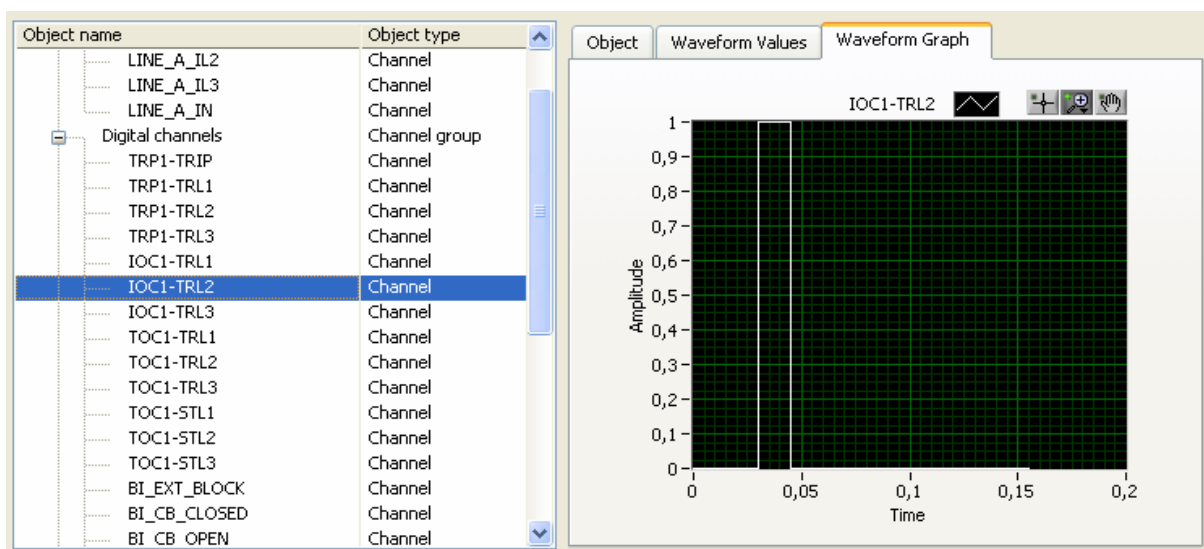
Ověření správné funkce vytvořeného DataPluginu bylo prováděno dvěma způsoby. Jednak je to testování ve *CreateDataPluginOutputFile.vi*, které simuluje architekturu DataPluginu volající *yourCodeHere.vi*, a jednak pomocí DIAdemu.

Testování pomocí *CreateDataPluginOutputFile.vi*

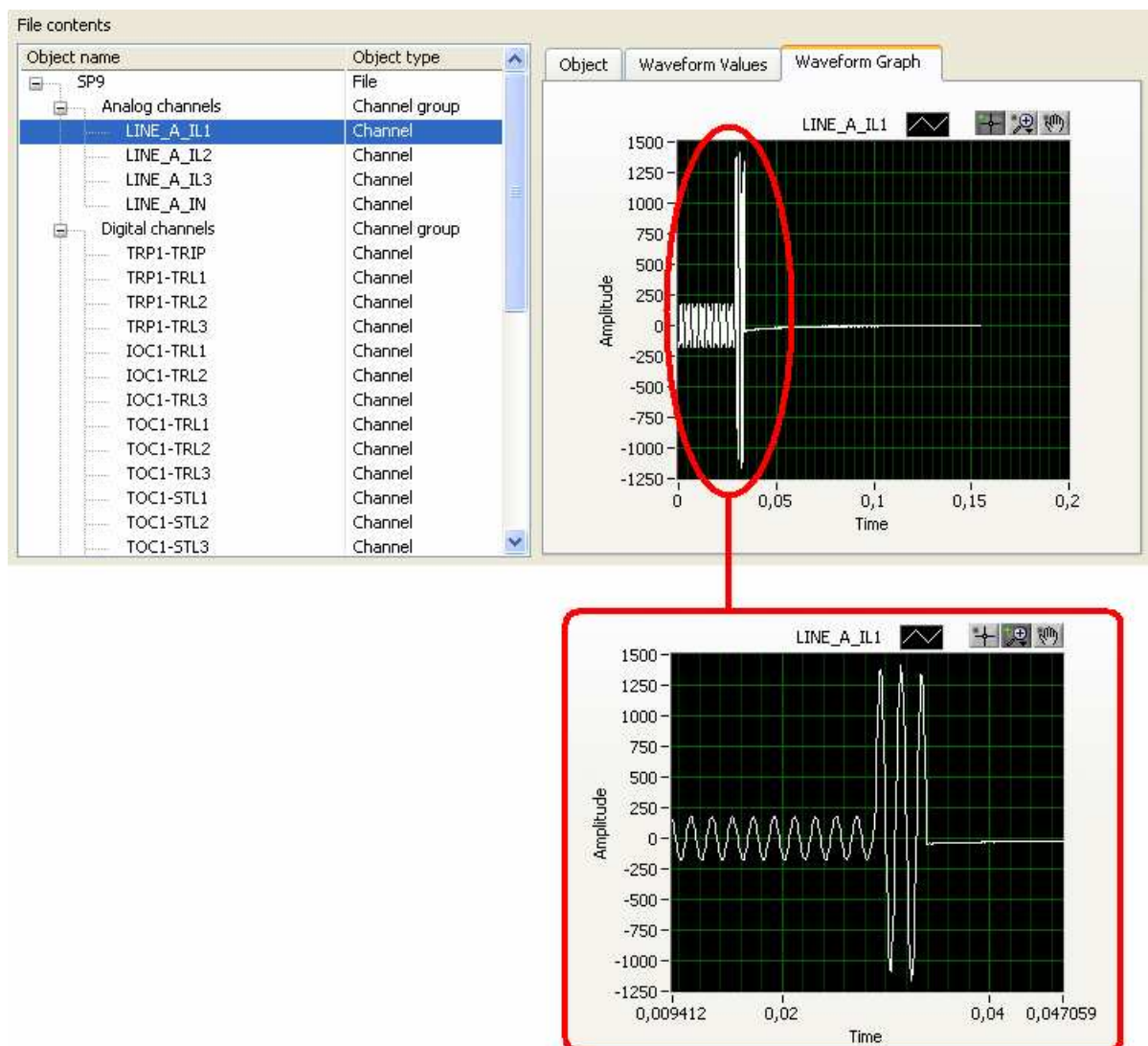
Následující obrázky jsou výsledkem testování v *CreateDataPluginOutputFile.vi*, které je součástí souborů vytvářejících strukturu projektu DataPluginu. Na obr. 30 jsou zobrazeny properties analogového kanálu, na obr. 31 jsou zachyceny hodnoty digitálního kanálu v grafu, a na obr. 32 jsou hodnoty analogového kanálu v grafu.



Obr. 30 Properties analogového kanálu



Obr. 31 Hodnoty digitálního kanálu zachycené v grafu

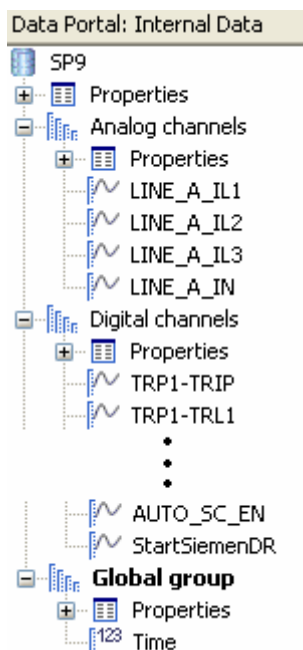


Obr. 32 Hodnoty analogového kanálu zachycené v grafu

Po spuštění testovacího souboru *CreateDataPluginOutputFile.vi* proběhne konverze dat z COMTRADE formátu do TDM bez problémů. Na základě výsledků tohoto testování, lze konstatovat, že DataPlugin funguje během tohoto testování správně. Na základě tohoto výsledku byl vytvořen instalátor *LabVIEW_DataPluginCOMTRADE*, který byl posléze zaveden do systému, aby mohlo být provedeno ověření správnosti ještě v DIAdemu.

Testování pomocí softwaru DIAdem

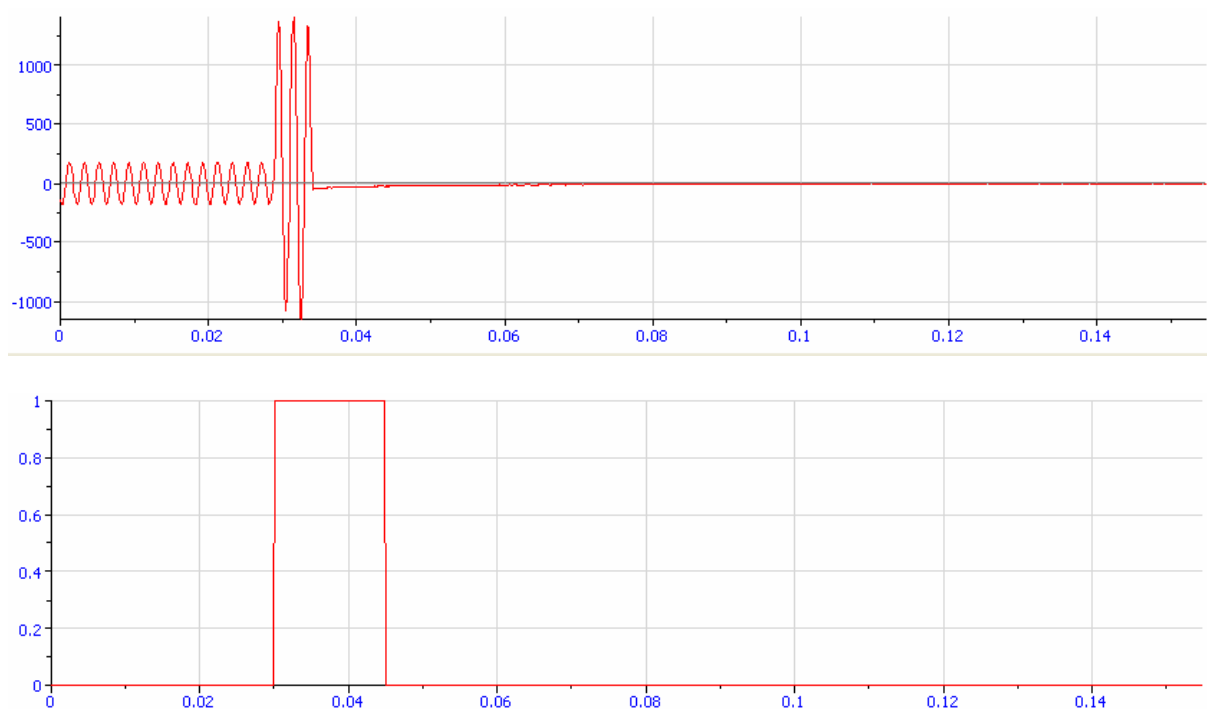
Pro ověření funkčnosti podpory formátu COMTRADE je potřeba vybrat příslušný datový soubor a ten přetáhnout myší do oblasti datového portálu. V tomto okamžiku je zahájena konverze dat z původního formátu do formátu TDM. Jakmile je převod dokončen, tak se do datového portálu načtou jednotlivé skupiny s příslušnými kanály (viz. obr. 33). V tomto portálu lze výběrem příslušné skupiny nebo kanálu zobrazit odpovídající properties. Pokud se vybraný kanál přetáhne do oblasti tabulky, tak se do ní načtou data z kanálu (viz. obr. 34). Při přetáhnutí kanálu do oblasti grafu dojde k vykreslení hodnot obsažených v kanálu (viz. obr. 35).



Obr. 33 Načtené skupiny a kanály do datového portálu DIAdemu

Selected Channels			
Name	Time	LINE_A_IL1	IOC1-TRL2
Number	32	1	10
Length	1549	1549	1549
Unit	microseconds		
Channel Contents			
298	297	1128.5761806001	0
299	298	807.877288799966	0
300	299	415.897637279937	0
301	300	34.4143680001143	0
302	301	-285.811326239...	1
303	302	-592.357309200...	1
304	303	-889.396323000...	1
305	304	-1078.58931108...	1
306	305	-1083.2352507601	1
307	306	-942.523503599...	1

Obr. 34 Tabulka s načtenými daty z kanálů v DIAdemu



Obr. 35 Graf analogového a digitálního kanálu zobrazeného v sekundách v DIAdemu

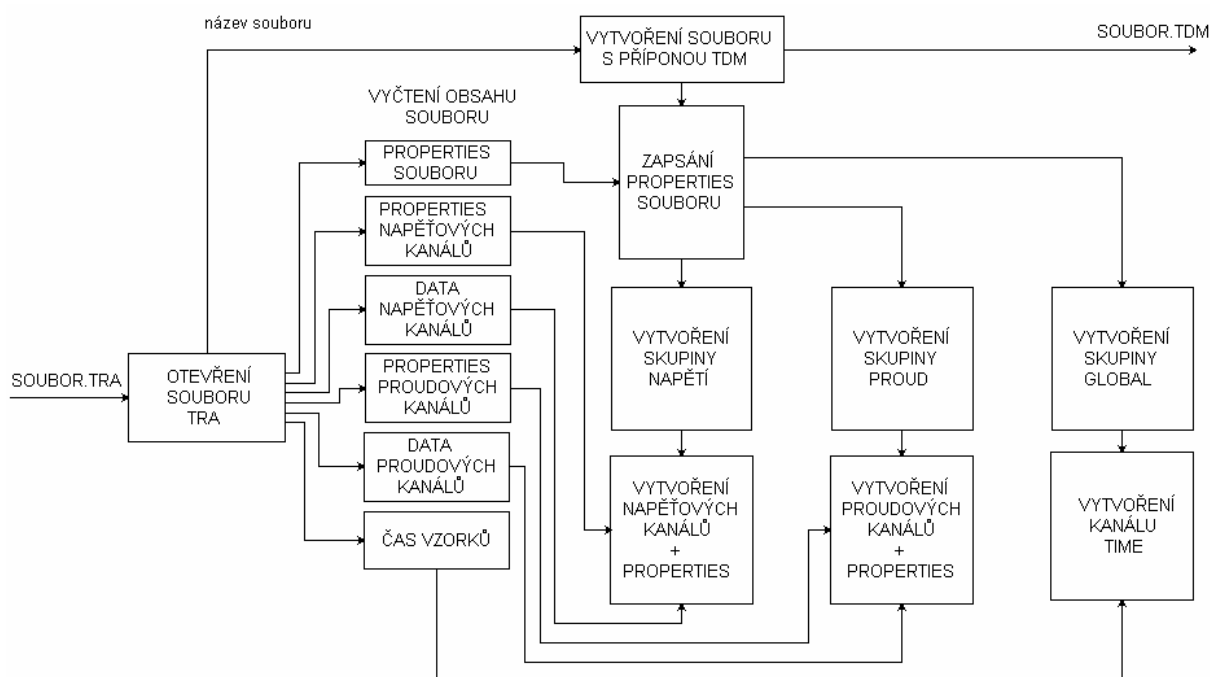
Ověření vytvořeného DataPluginu proběhlo i v tomto případě zcela bez problémů a navíc pokud se srovnají grafy z tohoto a předchozího způsobu testování, tak si lze povšimnout, že průběhy jsou totožné. Na základě výsledků obou způsobů testování lze říci, že DataPlugin byl navržen správně.

5.3 DataPlugin podporující souborové formáty TRA

Tento formát souboru se využívá pro zápis dat naměřených pomocí speciálního modulu transientního zapisovače analyzátoru napájecích sítí BK. Pro každou událost uloženou transientním zapisovačem jsou uvedeny následující informace: Time (čas), S/Rate, napětí U_x na jednotlivých fázích a proud I_x na jednotlivých fázích.

5.3.1 Návrh jádra DataPluginu one shot pro podporu *. TRA

Pro vytvoření jádra DataPluginu bylo využito subVI *READ TRA* získané od výrobce analyzátoru. Pomocí něj byly v jádru vyčteny veškeré informace uložené v souboru *. TRA. Tyto informace jsou nezbytné pro vytvoření DataPluginu. Vyčtené informace se dělí na informace týkající se pouze souboru (properties), informace skupin kanálů (properties), informace kanálů včetně dat. Tyto data byly zapsány do souboru TDM. Postup vytváření souboru TDM byl obdobný, jako je tomu v kapitole 5.2.3. Aby bylo možné data, které obsahuje kanál, zobrazovat v grafu, je potřeba nastavit všechny properties týkající se grafů. Mezi jednu z nejdůležitějších patří velikost inkrementu, což je v podstatě velikost časového kroku mezi dvěma následujícími vzorky. Blokové schéma jádra DataPluginu je v příloze na CD.



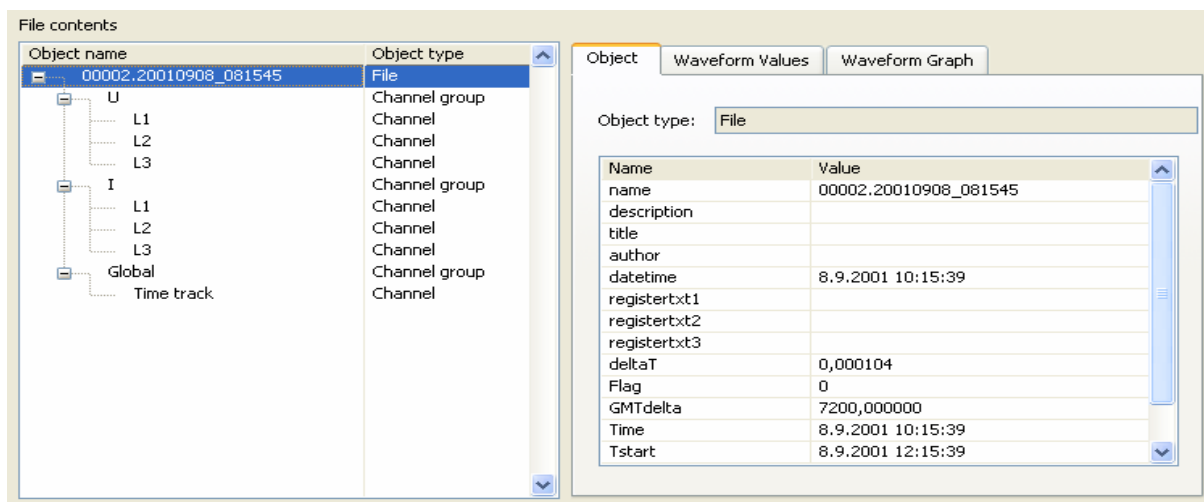
Obr. 36 Blokový diagram DataPluginu TRA

5.3.2 Testování DataPluginu podporujícího formát TRA

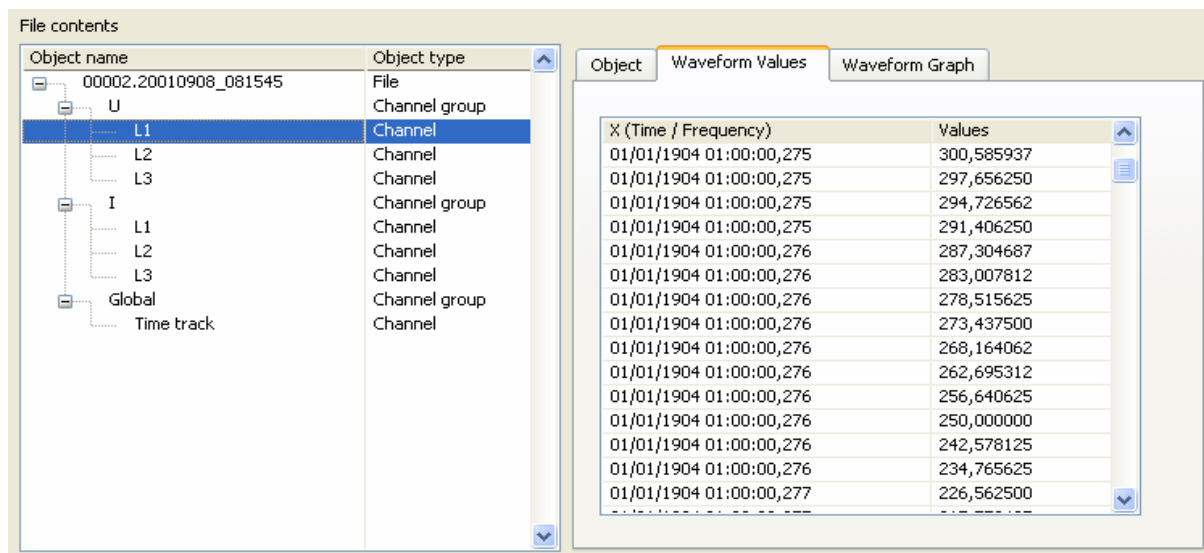
Ověření správné funkce vytvořeného DataPluginu bylo prováděno dvěma způsoby. Jednak je to testování ve *CreateDataPluginOutputFile.vi*, které simuluje architekturu DataPluginu volající *yourCodeHere.vi*, a jednak pomocí DIAdemu.

Testování pomocí *CreateDataPluginOutputFile.vi*

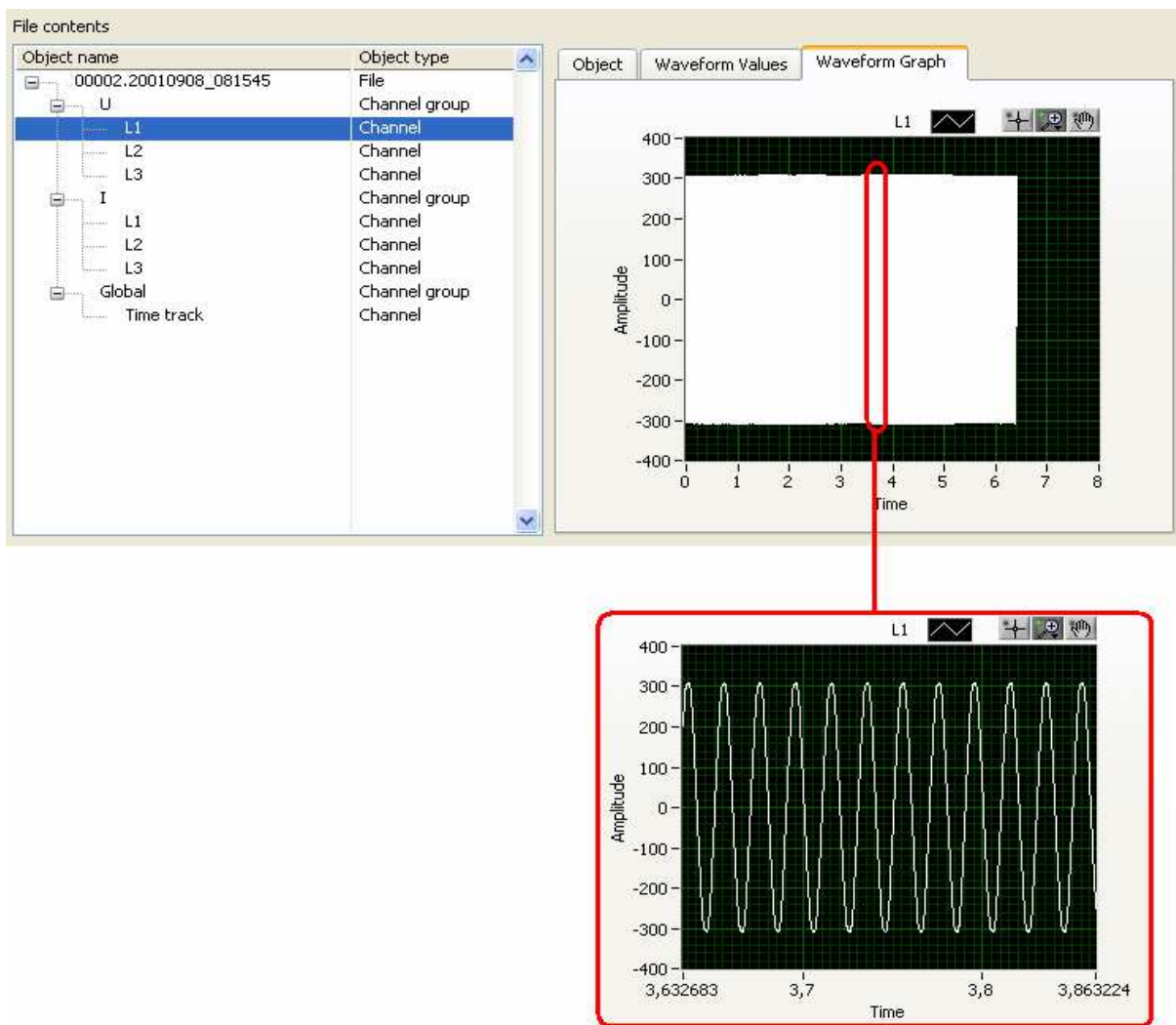
Následující obrázky jsou výsledkem testování v *CreateDataPluginOutputFile.vi*, které je součástí souborů vytvářejících strukturu projektu DataPluginu. Na obr. 37 jsou zobrazeny properties souboru, na obr. 38 jsou zachyceny hodnoty kanálu L1, který je ve skupině U, a na obr. 39 jsou hodnoty tohoto kanálu zobrazeny v grafu.



Obr. 37 Properties souboru



Obr. 38 Hodnoty kanálu L1 (skupina U)

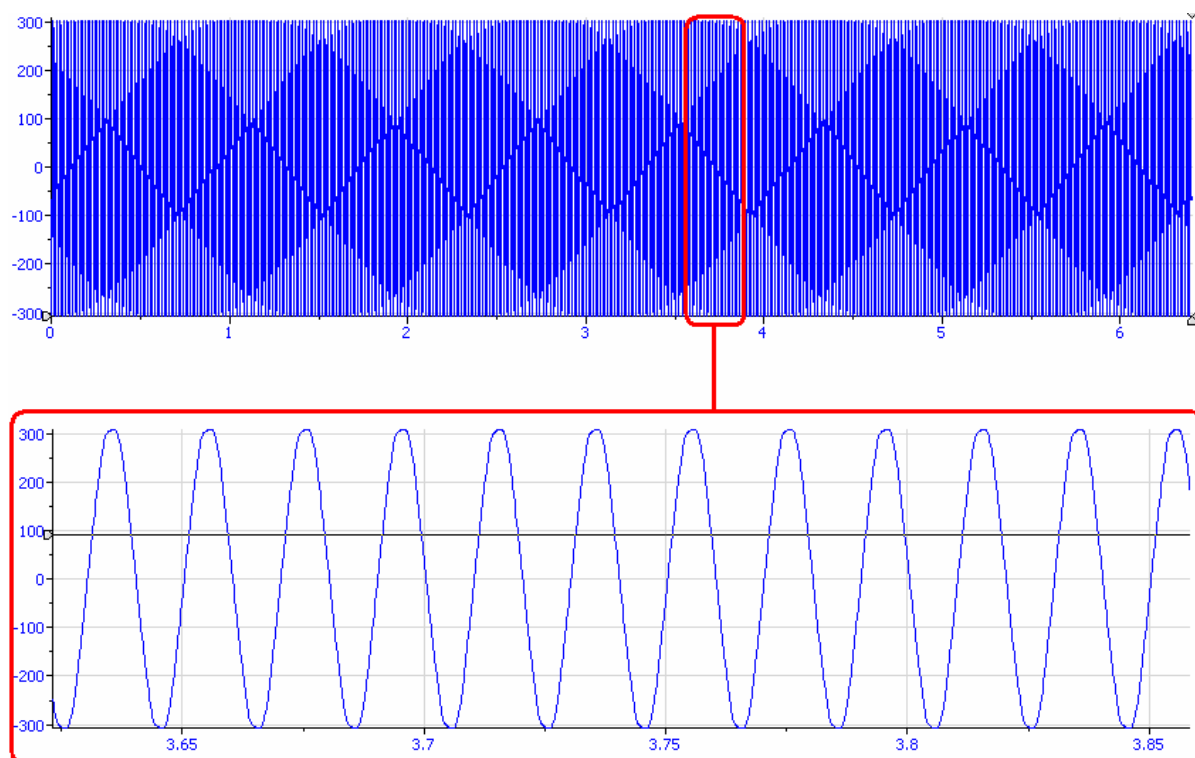


Obr. 39 Hodnoty kanálu L1 zachycené v grafu (skupina U)

Po spuštění testovacího souboru *CreateDataPluginOutputFile.vi* proběhne konverze dat z formátu TRA do formátu TDM bez problémů. Na základě výsledků tohoto testování, lze konstatovat, že DataPlugin funguje během tohoto testování správně. Na základě tohoto výsledku byl vytvořen instalátor *LabVIEW_DataPluginTRA*, který byl posléze zaveden do systému, aby mohlo být provedeno ověření správnosti ještě v DIAdemu.

Testování pomocí softwaru DIAdem

Pro ověření funkčnosti podpory formátu TRA je potřeba pouze najít příslušný datový soubor a ten přetáhnout myší do oblasti datového portálu. V tomto okamžiku je zahájena konverze dat z původního formátu do formátu TDM. Jakmile je převod dokončen, tak se do datového portálu načtou jednotlivé skupiny s příslušnými kanály. V tomto portálu lze výběrem příslušné skupiny nebo kanálu zobrazit odpovídající properties. Pokud se vybraný kanál přetáhne do oblasti tabulky, tak se do ní načtou data z kanálu. Při přetáhnutí kanálu do oblasti grafu dojde k vykreslení hodnot obsažených v kanále (viz. obr. 40).



Obr. 40 Graf kanálu L1 (skupina U) zobrazeného v sekundách v DIAdemu

Ověření vytvořeného DataPluginu proběhlo i v tomto případě zcela bez problémů. Na základě výsledků obou způsobů testování lze říci, že DataPlugin byl navržen správně. Grafy u obou způsobu testování mají naprosto totožné průběhy.

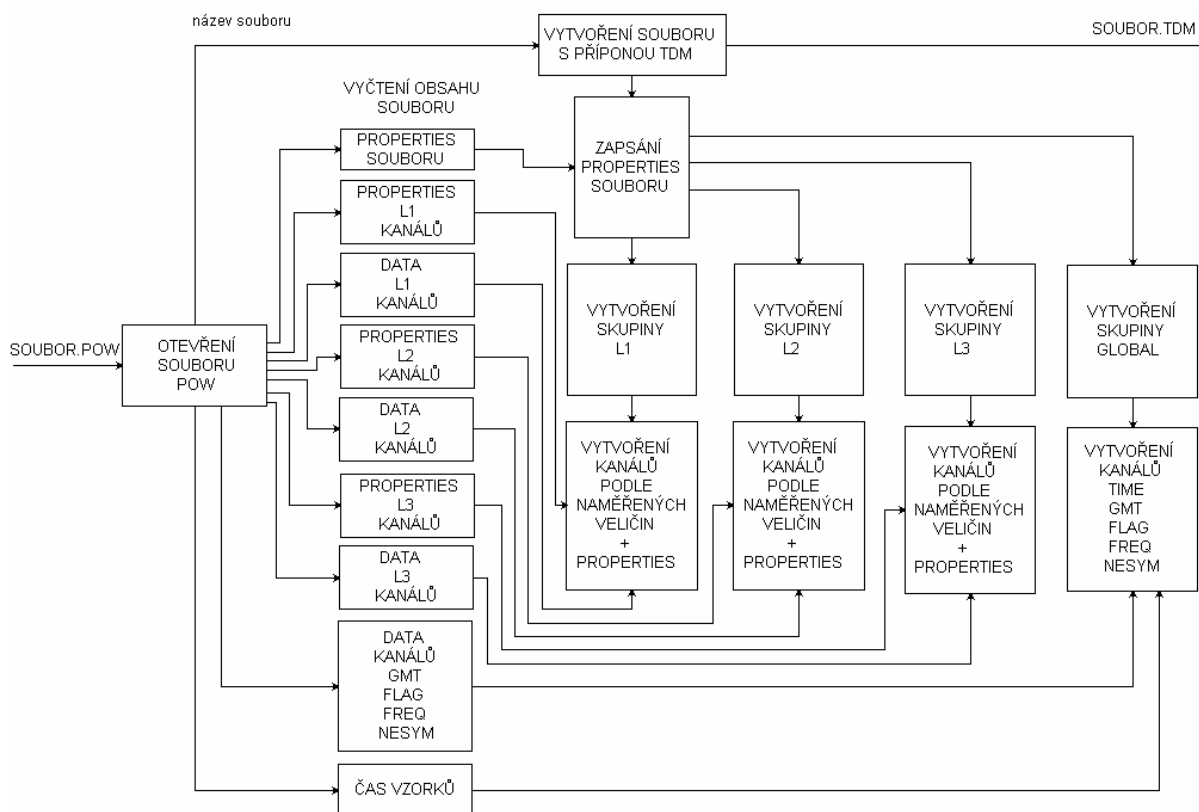
5.4 DataPlugin podporující souborové formáty POW

Tento formát se používá pro ukládání dat, týkajících se výkonu fází L1, L2, L3, analyzátozem napájecích sítí. Každá fáze obsahuje naměřené veličiny: THDu, THDi, U, I, P, S, Q, P1, Q1, cos fi, PF, res1, res2, res3, res4.

5.4.1 Návrh jádra DataPluginu one shot pro podporu *. POW

Pro vytvoření jádra DataPluginu bylo využito subVI *READ POW* získané od výrobce analyzátoru. Pomocí něj byly v jádru vyčteny veškeré informace uložené v souboru *. POW. Tyto informace jsou nezbytné pro vytvoření DataPluginu. Vyčtené informace se dělí na informace týkající se pouze souboru (properties), informace skupin kanálů (properties), informace kanálů včetně dat. Tyto data byly zapsány do souboru TDM. Postup vytváření souboru TDM byl obdobný, jako je tomu v kapitole 5.2.3. Aby bylo možné data, které obsahuje kanál, zobrazovat v grafu, je potřeba nastavit všechny properties týkající se grafů. Mezi jednu z nejdůležitějších patří velikost inkrementu, což je v podstatě velikost časového kroku mezi dvěma následujícími vzorky.

Blokové schéma jádra DataPluginu je v příloze na CD.



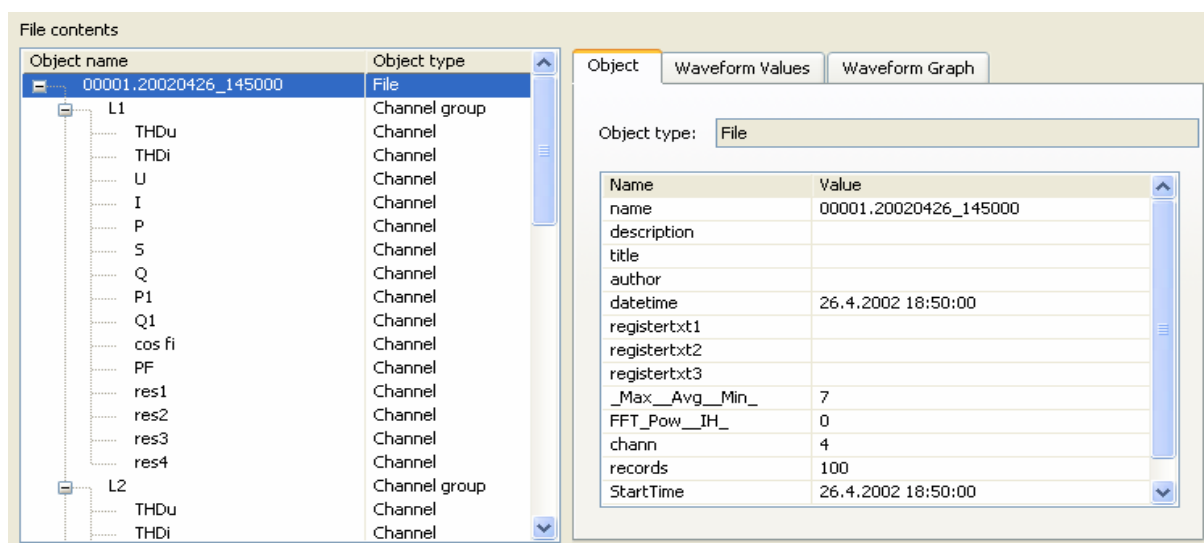
Obr. 41 Blokový diagram DataPluginu POW

5.4.2 Testování DataPluginu podporujícího formát POW

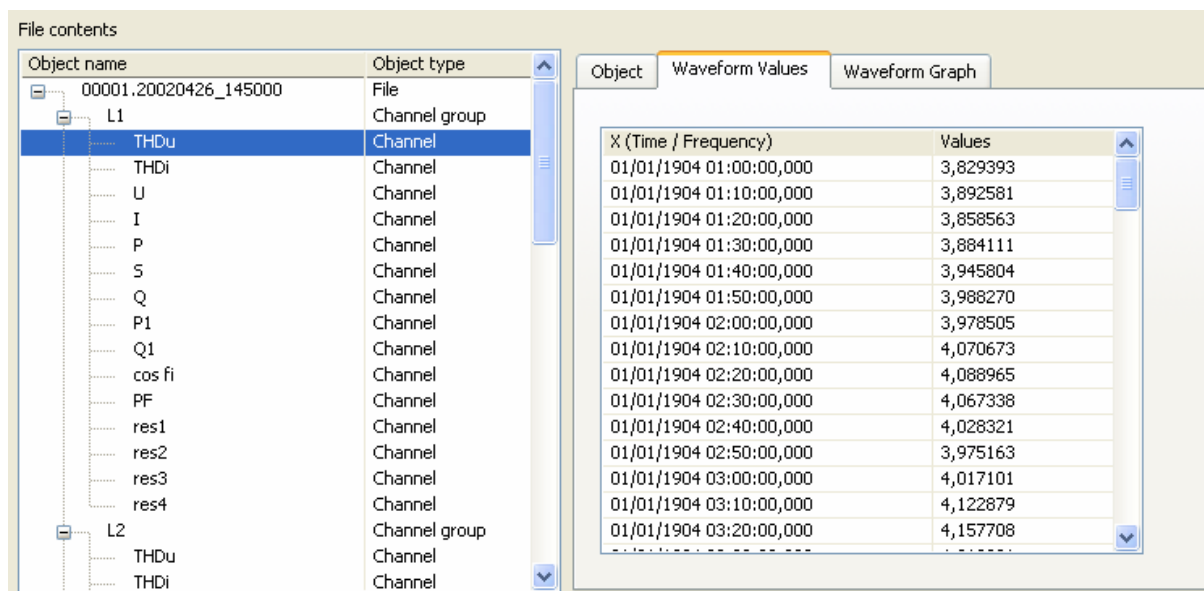
Ověření správné funkce vytvořeného DataPluginu bylo prováděno dvěma způsoby. Jednak je to testování ve *CreateDataPluginOutputFile.vi*, které simuluje architekturu DataPluginu volající *yourCodeHere.vi*, a jednak pomocí DIAdemu.

Testování pomocí *CreateDataPluginOutputFile.vi*

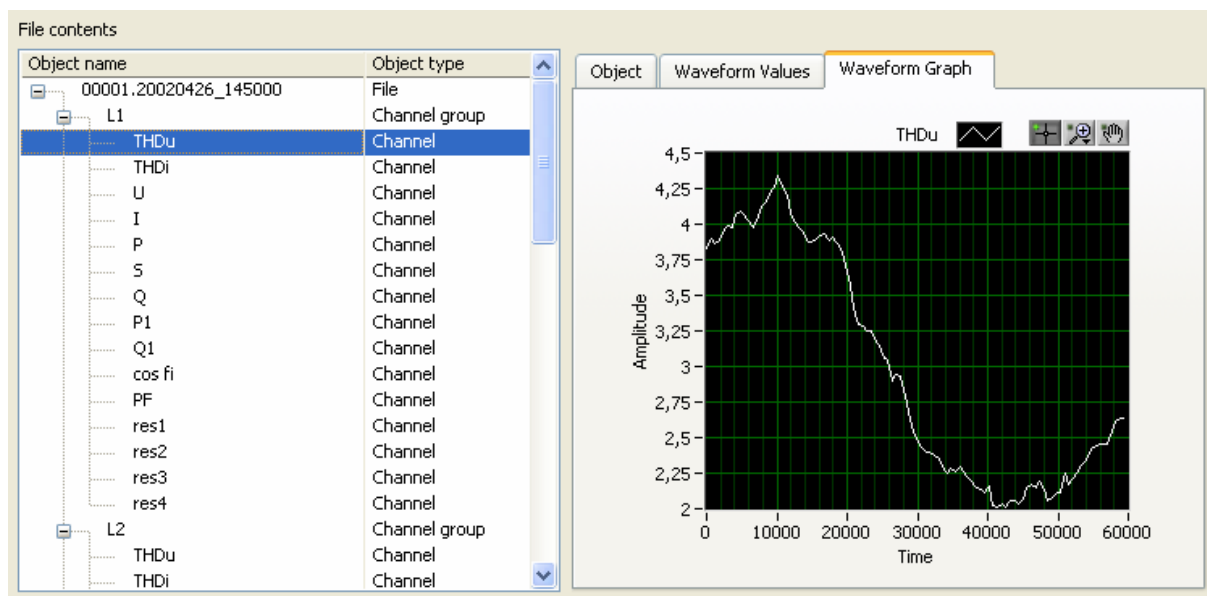
Následující obrázky jsou výsledkem testování v *CreateDataPluginOutputFile.vi*, které je součástí souborů vytvářejících strukturu projektu DataPluginu. Na obr. 42 jsou zobrazeny properties souboru, na obr. 43 jsou zachyceny hodnoty kanálu THDu, který je ve skupině L1, a na obr. 44 jsou hodnoty tohoto kanálu zobrazeny v grafu.



Obr. 42 Properties souboru



Obr. 43 Hodnoty kanálu THDu (skupina L1)

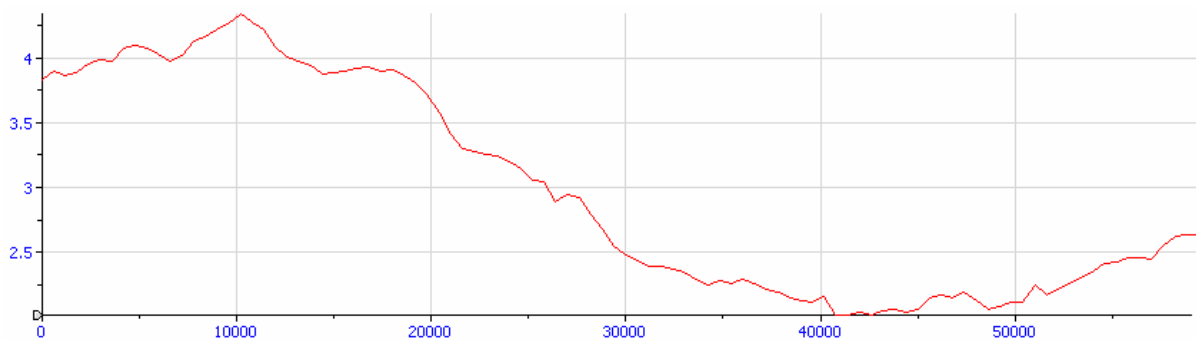


Obr. 44 Hodnoty kanálu THDu zachycené v grafu (skupina L1)

Po spuštění testovacího souboru *CreateDataPluginOutputFile.vi* proběhne konverze dat z formátu POW do formátu TDM bez problémů. Na základě výsledků tohoto testování, lze konstatovat, že DataPlugin funguje během tohoto testování správně. Na základě tohoto výsledku byl vytvořen instalátor *LabVIEW_DataPluginPOW*, který byl posléze zaveden do systému, aby mohlo být provedeno ověření správnosti ještě v DIAdemu.

Testování pomocí softwaru DIAdem

Pro ověření funkčnosti podpory formátu POW je potřeba pouze najít příslušný datový soubor a ten přetáhnout myší do oblasti datového portálu. V tomto okamžiku je zahájena konverze dat z původního formátu do formátu TDM. Jakmile je převod dokončen, tak se do datového portálu načtou jednotlivé skupiny s příslušnými kanály. V tomto portálu lze výběrem příslušné skupiny nebo kanálu zobrazit odpovídající properties. Pokud se vybraný kanál přetáhne do oblasti tabulky, tak se do ní načtou data z kanálu. Při přetáhnutí kanálu do oblasti grafu dojde k vykreslení hodnot obsažených v kanále (viz. obr. 45).



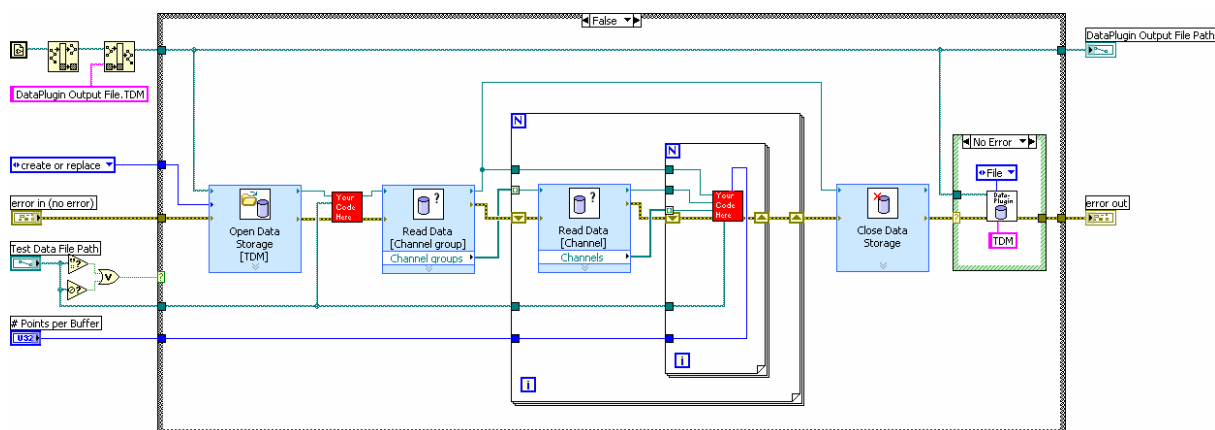
Obr. 45 Graf kanálu THDu (skupina L1) zobrazený v sekundách v DIAdemu

Ověření vytvořeného DataPluginu proběhlo i v tomto případě zcela bez problémů. Na základě výsledků obou způsobů testování lze říci, že DataPlugin byl navržen správně. Grafy u obou způsobu testování mají naprosto totožné průběhy.

6 DataPlugin on demand podporující formát COMTRADE

Jedná se o více komplikovaný a náročnější způsob pro vytváření DataPluginu, ale obecně se dá říci, že je vhodnější pro rozsáhlé datové soubory. On Demand DataPlugin obsahuje dvě volání, *yourCodeHere_meta.vi*, který je volán pouze jednou během provádění DataPluginu a pak *yourCodeHere_raw.vi*, který může být volán jednou, ale i vícekrát. Záleží to na tom, kdy jsou vyžadována data. *YourCodeHere_meta.vi* deklaruje datové kanály, které jsou specifické například datovým typem, názvem, velikostí (délkou), popisnými vlastnostmi (properties), ale neobsahují žádné data. *YourCodeHere_raw.vi* je volán individuálně pro každý datový kanál. [3]

Výše popsany princip tohoto typu DataPluginu je zobrazen na obr. 46.

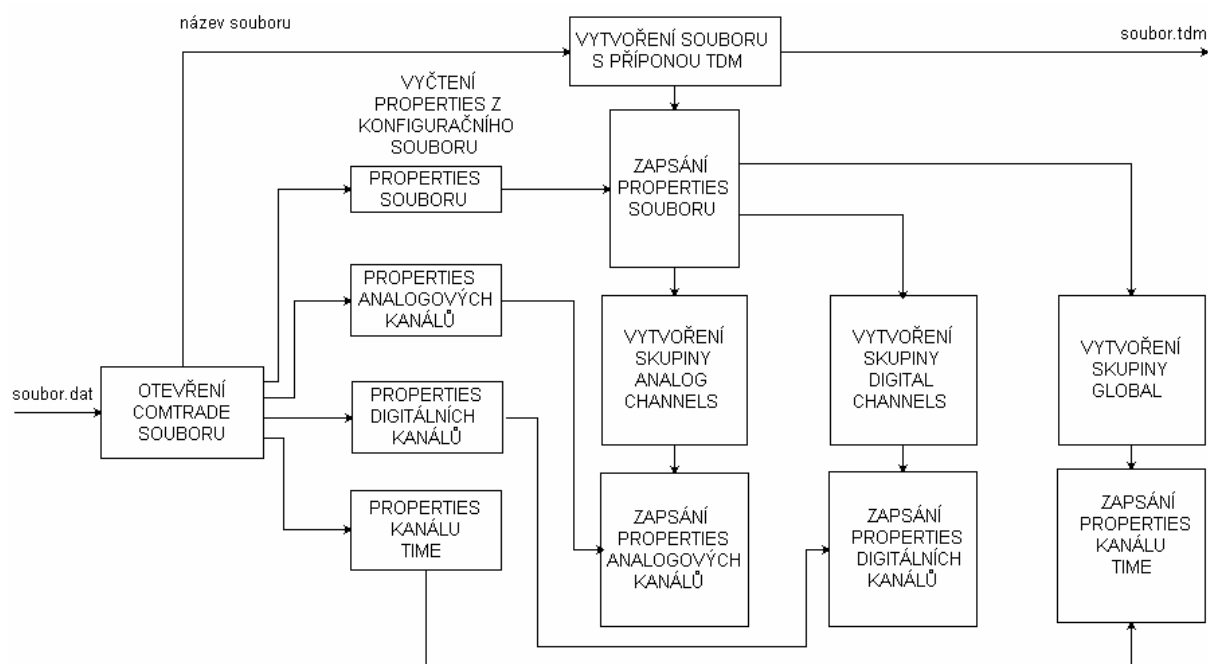


Obr. 46 Princip funkce DataPluginu typu on demand

6.1 Návrh jádra DataPluginu

6.1.1 Meta data

Jádro DataPluginu je tvořeno tedy dvěma VI. Nejprve se vytváří VI řešící meta data, což jsou v podstatě veškeré informace o souboru, skupinách a kanálech, avšak nesmí obsahovat žádné data kanálů, může obsahovat pouze properties. Pro vytvoření jádra s meta daty bylo využito subVI *Read Comtrade*, které je schopno vyčíst obsah souborů ve formátu COMTRADE, tzn. z konfiguračního a datového souboru. Z obsahu těchto souborů budou v rámci meta dat vyčteny properties souboru, skupin a kanálů. Nejprve byl vytvořen TDM soubor stejného názvu, jako měl soubor ve formátu COMTRADE. Po zapsání properties souboru následovalo vytvoření skupin kanálů (analog a digital) a přiřazení příslušejících properties. V neposlední řadě bylo potřeba vytvořit jak analogové tak digitální kanály, do kterých byly zapsány pouze properties, mezi nimiž je i index kanálu, který je pak využíván v rámci jádra pro raw data (to jsou data kanálů). Aby bylo možné data, které obsahuje kanál, zobrazovat v grafu, je potřeba nastavit všechny properties týkající se grafů. Mezi jednu z nejdůležitějších patří velikost inkrementu, což je v podstatě velikost časového kroku mezi dvěma následujícími vzorky. Navíc byla vytvořena skupina Global, do které byl zapsán kanál, který obsahuje časové razítko (respektive jen čas sejmutí vzorku v ms). Blokové schéma jádra DataPluginu je v příloze na CD.



Obr. 47 Blokový diagram části jádra s META daty

6.1.2 Raw data

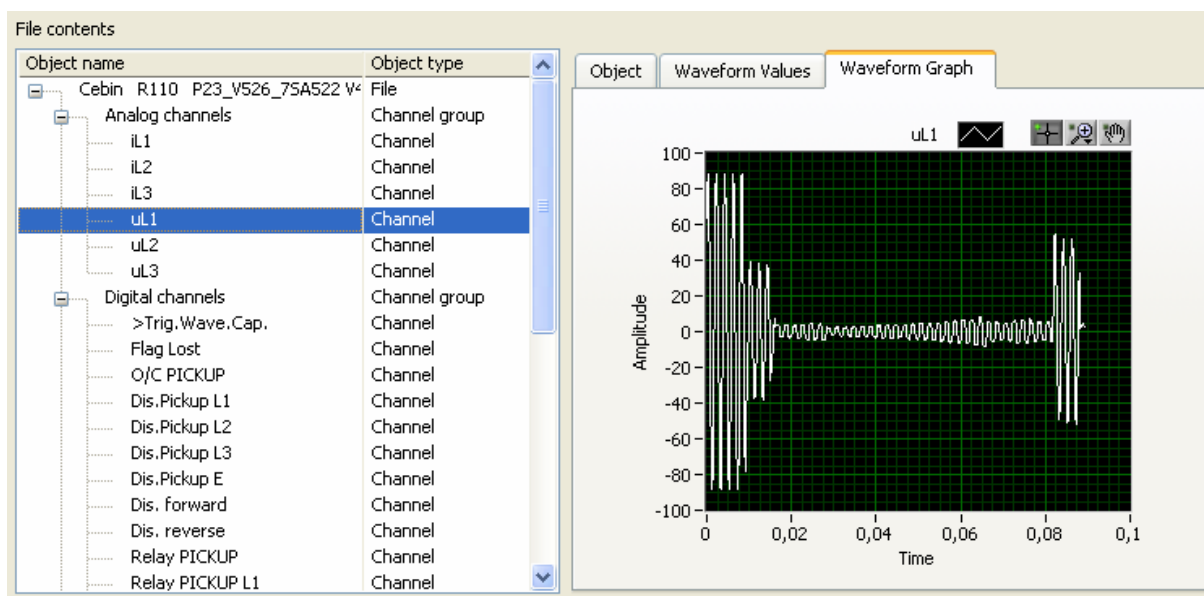
Tato část jádra DataPluginu je zaměřena na zapisování dat do kanálů a je propojena s částí, která řeší meta data. Data jsou zapisována tak, že se nejprve zjistí, o jakou skupinu kanálu se jedná (podle názvu skupiny), poté se na základě indexu kanálu vybere příslušný kanál a do něj se zapíše data. Pro vyčtení dat z COMTRADE souboru je zde opět využito subVI *Read Comtrade*. Blokové schéma jádra DataPluginu je v příloze na CD.

6.2 Testování DataPluginu podporujícího COMTRADE

Ověření správné funkce vytvořeného DataPluginu bylo prováděno dvěma způsoby. Jednak je to testování ve *CreateDataPluginOutputFile.vi*, které simuluje architekturu DataPluginu volající *yourCodeHere.vi*, a jednak pomocí DIAdemu.

Testování pomocí *CreateDataPluginOutputFile.vi*

Následující obrázek je výsledkem testování v *CreateDataPluginOutputFile.vi*, které je součástí souborů vytvářejících strukturu projektu DataPluginu. Na obr. 48 jsou zobrazeny hodnoty analogového kanálu v grafu.

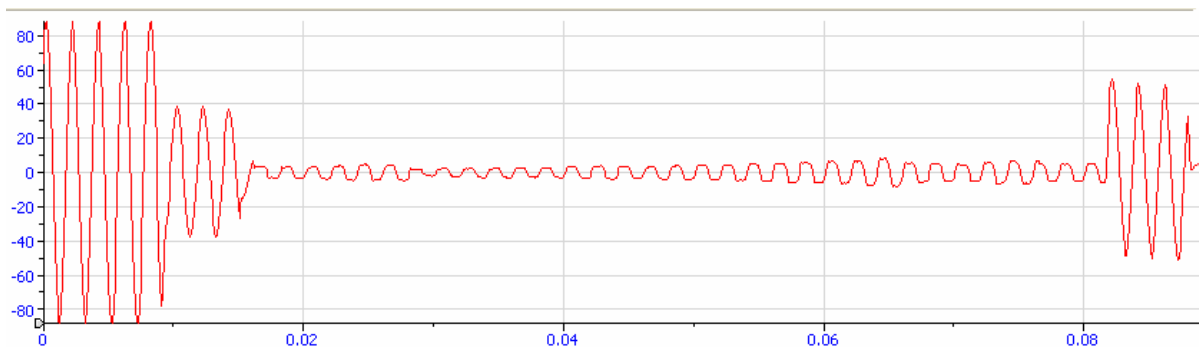


Obr. 48 Hodnoty analogového kanálu zachycené v grafu

Po spuštění testovacího souboru *CreateDataPluginOutputFile.vi* proběhne konverze dat z COMTRADE formátu do TDM bez problémů. Na základě výsledků tohoto testování, lze konstatovat, že DataPlugin funguje během tohoto testování správně. Na základě tohoto výsledku byl vytvořen instalátor *LabVIEW_DataPluginCOMTRADEOnDemand*, který byl posléze zaveden do systému, aby mohlo být provedeno ověření správnosti ještě v DIAdemu.

Testování pomocí softwaru DIAdem

Pro ověření funkčnosti podpory formátu COMTRADE je potřeba pouze najít příslušný datový soubor a ten přetáhnout myší do oblasti datového portálu. V tomto okamžiku je zahájena konverze dat z původního formátu do formátu TDM. Jakmile je převod dokončen, tak se do datového portálu načtou jednotlivé skupiny s příslušnými kanály. V tomto portálu lze výběrem příslušné skupiny nebo kanálu zobrazit odpovídající properties. Pokud se vybraný kanál přetáhne do oblasti tabulky, tak se do ní načtou data z kanálu. Při přetáhnutí kanálu do oblasti grafu dojde k vykreslení hodnot obsažených v kanále (viz. obr. 49).



Obr. 49 Graf analogového kanálu zobrazeného v sekundách v DIAdemu

Ověření vytvořeného DataPluginu proběhlo i v tomto případě zcela bez problémů a navíc pokud se srovnají grafy z tohoto a předchozího způsobu testování, tak si lze povšimnout, že průběhy jsou totožné. Na základě výsledků obou způsobů testování lze říci, že DataPlugin byl navržen správně.

7 Závěr

Výsledkem této diplomové práce je softwarová podpora datových formátů COMTRADE, TRA a POW, které se používají pro ukládání naměřených dat v elektrotechnice. Jedná se o podporu datových formátů v prostředí softwaru DIAdem, který je určen pro analýzu a zpracování naměřených dat. Softwarová podpora byla vytvářena v prostředí produktu LabVIEW pomocí DataPluginů, které obsahují strukturu souborů vytvořených v LabVIEW. Pomocí této struktury je pak vytvářena DLL knihovna a následně instalátor podpory datového formátu.

Nejprve byly vytvářeny podpory pomocí DataPluginu typu one shot. Prvním datovým formátem, pro který byla vytvářena tato podpora, byl COMTRADE. K tomu, aby bylo možné vyčíst informace z konfiguračního a datového souboru tohoto formátu, muselo být vytvořeno VI, které zajišťuje přečtení dat z COMTRADE formátu. Zároveň bylo vytvořeno i VI, které umožňuje data do COMTRADE zapisovat. Vytvořené VI pro čtení ze souborů se stalo velmi důležitou součástí vytvářeného jádra DataPluginu. Díky němu bylo možné vyčíst ze souboru data, které pak byly při konverzi zapisovány do TDM souborů.

Další vytvářené podpory pomocí typu one shot byly pro datové formáty TRA a POW, což jsou formáty určené pro ukládání dat týkajících se kvality napájecích sítí. Jsou to proprietární datové formáty výrobce analyzátoru. Čtečky těchto formátů byly poskytnuty za účelem této diplomové práce.

Všechny podpory typu one shot byly úspěšně otestovány, jak v rámci struktury DataPluginu, tak i v samotnému DIAdemu, pro který byly vytvářeny.

Pro typ DataPluginu on demand byl vybrán datový formát COMTRADE, aby bylo možné porovnat oba způsoby vytváření DataPluginu na stejném datovém formátu. Podpora tímto způsobem byla také vytvořena a její funkčnost náležitě ověřena.

Co se týká zhodnocení obou metod, tak se dá říci, že jednodušším a rychlejším způsobem vytváření podpory je bezesporu one shot, protože zde je jádro tvořeno pouze jedním VI, které se upravuje programátorem. Tím pádem se stává přehlednějším a lépe se v něm provádí korekce. Naproti tomu je nutno podotknout, že on demand je podporou, která je vhodná pro soubory buďto s velkým počtem kanálů nebo s kanály, které obsahují velké množství záznamů. Ale tím, že programátor musí v rámci tohoto typu upravovat jádro se dvěma VI, které se na sebe různě odkazují, se stává tvorba podpory zejména složitějších datových typů komplikovanější nežli u one shot.

Použitá literatura

- 1) *National Instruments : NI Developer Zone* [online]. 2007-12-03 [cit. 2010-04-01]. National Instruments Technical Data Management Overview. Dostupné z WWW: <<http://zone.ni.com/devzone/cda/tut/p/id/3676>>.
- 2) *National Instruments : NI Developer Zone* [online]. 2009-09-05 [cit. 2010-03-05]. Introduction to LabVIEW TDM Streaming VIs. Dostupné z WWW: <http://zone.ni.com/devzone/cda/tut/p/id/3539>.
- 3) IEEE Std C37.111-1999. *IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems*. 3 Park Avenue, New York, NY 10016-5997, USA : The Institute of Electrical and Electronics Engineers, Inc., 1999-10-15. 55 s.
- 4) ŽÍDEK, Jan. Měřicí systémy a jejich programování. Ostrava: VŠB-TUO, [200?]. SW nástroje managementu naměřených dat, s. 249-252.
- 5) *National Instruments : Manuals* [online]. 2008-08-12 [cit. 2010-05-03]. LabVIEW DataPlugin SDK. Dostupné z WWW: <<http://www.ni.com/pdf/manuals/372601a.pdf>>.

Seznam příloh

Příloha č. 1 - Struktura konfiguračního souboru	I
Příloha č. 2 - Struktura informačního souboru	II
Příloha č. 3 - Přiložené CD	VII